



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΜΙΚΡΟΗΛΕΚΤΡΟΝΙΚΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση συστημάτων σε ψηφίδα (SoC)

**για τη στοχαστική προσομοίωση μεγάλων δικτύων βιοχημικών
αντιδράσεων**

Ορσαλία Γεωργία Χαζάπη

Επιβλέπων: Ηλίας Μανωλάκος, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΑΘΗΝΑ

ΜΑΙΟΣ 2012

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση συστημάτων σε ψηφίδα (SoC)
για τη στοχαστική προσομοίωση μεγάλων δικτύων βιοχημικών αντιδράσεων

ΟΡΣΑΛΙΑ ΓΕΩΡΓΙΑ ΧΑΖΑΠΗ

ΑΜ: MM158

ΕΠΙΒΛΕΠΩΝ:

Ηλίας Μανωλάκος, Αναπληρωτής Καθηγητής ΕΚΠΑ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Ηλίας Μανωλάκος, Αναπληρωτής Καθηγητής ΕΚΠΑ

Αγγελική Αραπογιάννη, Καθηγήτρια ΕΚΠΑ

ΜΑΙΟΣ 2012

ΠΕΡΙΛΗΨΗ

Η παρούσα Διπλωματική Εργασία, αφορά την υλοποίηση ενός ψηφιακού συστήματος (SoC) ειδικού σκοπού με στόχο την στοχαστική προσομοίωση μεγάλων δικτύων βιοχημικών αντιδράσεων. Η υλοποίηση έγινε σε FPGAs τόσο της Xilinx όσο και της Altera. Η εργασία υλοποιεί στο υλικό τους δημοφιλείς αλγορίθμους στοχαστικής προσομοίωσης του Gillespie που στοχεύουν στην περιγραφή της εξέλιξης των δυναμικών αλληλεπιδράσεων των μορίων ενός βιολογικού συστήματος. Με την παρούσα εργασία προτείνονται διαφορετικές σχεδιάσεις μικροϋπολογιστών συστημάτων για πολλαπλή παράλληλη επεξεργασία των αλγορίθμων αλλά ταυτόχρονα και για την εκτέλεση ανεξάρτητων πολλαπλών προσομοιώσεων που μπορεί να εκτελούνται στο σύστημα παράλληλα. Για τον λόγο αυτό κατασκευάσαμε ψηφιακό σύστημα σε ψηφίδα (SoC) με παράλληλους επεξεργαστές το οποίο έχει τη δυνατότητα να προσομοιώνει δύο από τους πιο γνωστούς αλγορίθμους για στοχαστική προσομοίωση τους (First Reaction Method [1] και Next Reaction Method [2]) με μια ψηφιακή υλοποίηση. Η συχνότητα λειτουργίας στα FPGA μπορεί να ανέρχεται στα 266-320 MHz με εσωτερικές καθυστερήσεις που κυμαίνονται από 55-85 κύκλους ανά αντίδραση. Επιπλέον σε ένα SoC μπορούν να προσομοιωθούν μέχρι 16K αντιδράσεις. Επιπροσθέτως έγινε μελέτη για την διεπαφή όλου του ψηφιακού συστήματος με τον επεξεργαστή LEON. Χρησιμοποιήθηκε το ερευνητικό πρόγραμμα SysPy [3] για τη δημιουργία ενός εργαλείου επικοινωνίας με το χρήστη με στόχο την αυτοματοποίηση των λειτουργιών της σύνθεσης του ψηφιακού συστήματος, ανάλογα με τις απαιτήσεις του βιοχημικού μοντέλου που προσομοιώνεται καθώς και για την μεταφορά αποτελεσμάτων στον Η/Υ.

Στην παρούσα εργασία αποδείχθηκε η ικανότητα του αλγορίθμου First Reaction Method SSA να παραλληλοποιείται και να προσομοιώνει αποδοτικά μεγάλα δίκτυα βιοχημικών αντιδράσεων. Επίσης αποδείχθηκε σε σχέση με την μέχρι τώρα βιβλιογραφία, η δυνατότητα γραμμικής αύξησης της απόδοσης του συστήματος στοχαστικών προσομοιώσεων με τον αριθμό των παράλληλων επεξεργασιών για μοντέλα με χιλιάδες αντιδράσεις. Επιπλέον στην παρούσα σχεδίαση επιτεύχθηκε μείωση των εσωτερικών καθυστερήσεων και αύξηση της συχνότητας λειτουργίας, καθιστώντας το σύστημα μας ένα από τα πλέον πιο αποδοτικά. Κατά την ερευνητική αυτή πορεία βρέθηκε τρόπος για την παραμετροποίηση της περιγραφής του συστήματος σε VHDL ως προς το αριθμό των βιοχημικών αντιδράσεων αλλά και ως προς τη τάξη αυτών. Με αυτό τον τρόπο καθίσταται δυνατή η προσομοίωση πολύπλοκων αντιδράσεων ανωτέρας τάξης (μεγαλύτερης του 2)

που με τα μέχρι τώρα δεδομένα δεν ήταν εφικτή στο υλικό. Επιπροσθέτως ερευνητικά έγινε και η μελέτη παραλληλοποίησης του αλγορίθμου NRM ώστε να γίνει αποδοτικότερος για μεγάλα συστήματα αντιδράσεων καθώς επίσης μελετήθηκαν και σχεδιάστηκαν ειδικοί ταξινομητές εύρεσης του μικρότερου χρόνου αντίδρασης με βάση την ανάλυση του αλγορίθμου NRM οι οποίοι επιταχύνουν σημαντικά την όλη διαδικασία.

Κλείνοντας, η παρούσα εργασία προσπάθησε να συμβάλει στο έργο της Συστημικής Βιολογίας ώστε να μπορούν να προσομοιωθούν μεγάλα συστήματα βιομοριακών αντιδράσεων που αφορούν τη μελέτη βιοχημικών συστημάτων όπως ιών, κυττάρων, και γενικά πολύπλοκων μηχανισμών για την γρήγορη εξέλιξη της βιολογίας και της βιοπληροφορικής.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Στοχαστική Προσομοίωση, Αλγόριθμοι Gillespie Direct Method, First Reaction Method, Gibson and Brucks Next Reaction Method, Σχεδίαση Ψηφιακών συστημάτων, Συστημική Βιολογία, Mass action Kinetics, Δίκτυα Βιοχημικών αντιδράσεων, Επεξεργαστής Leon, SysPy ,

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Ενσωματωμένα Συστήματα, Δίκτυα Βιοχημικών αντιδράσεων , Συστημική Βιολογία , SSA, DM,FRM, NRM, συστημάτων σε ψηφίδα (SoC),SBML , VHDL, SysPy, FPGAs,.

ABSTRACT

The rapid advances in the field of systems biology creates new challenges for computer science by requiring the creation of algorithms and associated computing systems that can simulate efficiently biochemical reaction networks with thousands of molecular species and/or reactions in a reasonable amount of time. The achievements in technology enables scientists to tackle important fundamental questions in biology involving genomics, proteomics and the investigation of the dynamics of molecular species expression in living organisms such as bacteria and viruses through studying their biochemical pathways, which will result in establishing a better understanding of mechanisms involved in detrimental diseases. To that end the Stochastic Simulation Algorithms developed by d. Gillespie have played a key role in the investigation of biological systems' dynamic behavior.

This thesis presents the design of scalable reconfigurable parametric Systems on Chip (SoC) that implement Gillespie's Stochastic Simulation Algorithms (SSA), namely the First Reaction Method [1] and Gibson and Bruck's Next Reaction Method [2]. We have proved that our architectures of scalable parametric SoCs can deliver performance (Mega reactions/sec) which is increasing linearly with the number of processors, when simulating bimolecular networks with up to 16K reactions in a single FPGA. This is important because it contributes to the constant progress of biology by enabling realistic simulation of increasingly larger reaction networks (e.g. large metabolic networks or whole species networks). In order to do so, we have designed and verified various scalable parametric FRM/NRM SoC architectures with $N=1,2,4$ Processing Elements for Xilinx Virtex 5 - 7 FPGAs, reaching clock frequencies as high as 320 MHz and delivering throughput as high as 2.6-3.1 Mega Reaction cycles/sec.

Furthermore we present a new approach which simulates biochemical reaction networks of different reaction orders with multiple products. Using it we are able to perform either one parallel simulation of a reaction network in parallel, or N different independent simulations of reaction networks within the same SoC with N Processing Elements, for both SSA algorithms (FRM, NRM). Moreover in order to be able to generate parallel architectures that can accelerate the stochastic simulation of biomodels of any complexity we have adopted several techniques for parallelizing and parameterising the SoC designs to the maximum extent.

SUBJECT AREA: Stochastic Simulation, Chemical Master Equations, Markovian methods for solving the CME, Gillespie's Direct method, Gillespie's First Reaction Method, Gibson and Brucks Next reaction Method, SBML, Biomodels Database, Systems Biology, FPGA's Xilinx and Altera, SoCs, Leon processor, SysPy

KEYWORDS: Embedded Systems, Reaction Networks, Systems Biology, SSA, DM,FRM, NRM, SoC,FPGAs, SBML, VHDL,SysPy,FPGAs

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1	
1.1 Εισαγωγή	23
1.2 Εισαγωγή στα CPLDs και FPGAs	27
1.3 Στοχαστικές προσομοιώσεις βιοχημικών αντιδράσεων	30
1.4 Ανάγκη για υλοποιήσεις σε επίπεδο υλικού και η σύγκριση με τις αντίστοιχες εφαρμογές σε επίπεδο λογισμικού	32
1.5 Κίνητρα για τη δική μας εργασία	36
1.6 Γενική περιγραφή της αρχιτεκτονικής των SoCs FRM-NRM + LEON σε Xilinx και σε ALTERA	38
1.7 Τι θα δούμε στην παρούσα εργασία	40
ΚΕΦΑΛΑΙΟ 2	
2.1 Εισαγωγή στη φυσική-βιοχημεία	41
2.2 Εισαγωγή στη στοχαστική προσομοίωση	43
2.3 Περιγραφή Αλγορίθμων Στοχαστικών Προσομοιώσεων (Αλγόριθμοι Gillespie)	44
2.4 Περιγραφή του Αλγορίθμου του Gillespie Direct Method	45
2.5 Περιγραφή FRM Αλγόριθμοι, υπολογιστική πολυπλοκότητα	48
2.6 Ανάλυση NRM Αλγόριθμοι, κίνητρα για την δημιουργία του αλγορίθμου και υπολογιστική πολυπλοκότητα	49
2.7 Γενική περιγραφή των μεθόδων και των εργαλείων προσομοίωσης Simulation tools (Copasi, iBioSim)	51
ΚΕΦΑΛΑΙΟ 3	
3.1 Εισαγωγή στην αρχιτεκτονική των FRM SoC	55
3.2 Γενική περιγραφή του FRM4X και όλων των υπό-μονάδων	56
3.3 Ανάλυση τρόπων λειτουργίας για N ξεχωριστές προσομοιώσεις και μία N παράλληλη προσομοίωση	58
3.4 Ανάλυση μίας N παράλληλης προσομοίωσης	64
3.5 Περιγραφή N παράλληλων προσομοιώσεων	67
3.6 Οργάνωση μνήμης	69
3.7 Μονάδα PE	71
3.8 Στατιστικά στοιχεία προσομοιώσεων σε Xilinx και Altera για τα SoCs FRM1X, FRM2X, FRM4X, FRM8X	76
ΚΕΦΑΛΑΙΟ 4	
4.1 Γενική περιγραφή του FRM4X-NRM4X SoC	83
4.2 Περιγραφή όλων των υπομονάδων FRM4X-NRM4X SoC	85
4.3 Περιγραφή και οργάνωση επιπλέον μνημών για την υλοποίηση του NRM	88
4.4 Μέθοδοι ταξινόμησης και εύρεσης ελάχιστης τιμής χρόνου	89
4.5 Στατιστικά στοιχεία προσομοίωσης για την υλοποίηση του FRM4X_NRM4X αλγορίθμου	92
4.6 Μονάδα στατικών στοιχείων	93
ΚΕΦΑΛΑΙΟ 5	
5.1 Εισαγωγή του επεξεργαστή LEON στα SoCs και διασύνδεση και προγραμματισμός μέσω του SysPy	97
5.2 Ένωση των SoCs με το AMBA Bus του Leon	99
5.3 Σύστημα επικοινωνίας χρήστη	100

5.4 Προγραμματισμός συστήματος SysPy με Leon και SSA SoCs	102
5.5 Στατιστικά στοιχεία για υλοποίηση σε Xilinx των SoCs FRM1X, FRM2X, FRM4X και FRM4X-NRM4X μετά την εισαγωγή του επεξεργαστή Leon (Users Interface) ...	104
Αποτελέσματα και Στατιστικά Στοιχεία Προσομοίωσης	107

ΚΕΦΑΛΑΙΟ 6

6.1 Επίδοση συστημάτων και Στατιστικά στοιχεία Προσομοίωσης	107
6.2 Αποτελέσματα επιδόσεων για τα SoCs FRM1X-2X-4X και FRM4X+NRM4X σε σχέση με υλοποιήσεις του FRM αλγόριθμου σε CPUs της Intel.....	107

ΚΕΦΑΛΑΙΟ 7

7.1 Τρόπος προγραμματισμού και ελέγχου του συστήματος	113
7.2 Biomodels που χρησιμοποιήθηκαν για το testing και η συνοπτική ανάλυση τους	114
7.3 Συγκρίσεις προσομοιώσεων των SOC's με MATLAB, Copasi, Ibiosim tool	116
7.4 Περιγραφή μοντέλου ενζυματικών αντιδράσεων	120
7.5 Προσομοίωση Biomodel 188 και 189	122
7.6 Μοντέλο περιγραφής της λειτουργίας της πρωτεΐνης A _{syn} στην νόσο του Parkinson	128
7.7 Μοντέλο περιγραφής της λειτουργίας του ενζύμου UCHL1 στην νόσο του Parkinson	131
7.8 Προσομοίωση μοντέλων στις μικρότερης κλίμακας προσομοιωτές FRM1X/FRM2X ..	134
7.8.1 Προσομοίωση μοντέλου E-Coli.	134
7.8.2 Προσομοίωση του Autoregulatory Gene Network Biomodel.....	137
7.9 Αποτελέσματα και συμπεράσματα της διαδικασίας των πολλαπλών προσομοιώσεων.	141

ΚΕΦΑΛΑΙΟ 8

8. Γενικά συμπεράσματα και μελλοντικές βελτιώσεις	143
---	-----

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Περιγραφή της ανάλυσης βιοχημικών αντιδράσεων μέσα από τη συστημική βιολογία.....	25
Σχήμα 2: Μοντέλο μελέτης βιοχημικών αντιδράσεων στο κύτταρο.	25
Σχήμα 3: Δομή μιας πολύπλοκης διάταξης προγραμματιζόμενης λογικής (CPLD).....	28
Σχήμα 4: Γενική δομή μιας διάταξης FPGA.....	28
Σχήμα 5: Περιγραφή Μαρκοβιανής Μεθόδου για προσομοίωση βιοχημικών αντιδράσεων.....	30
Σχήμα 6: Στιγμιότυπο στοχαστικής προσομοίωσης ενός δικτυώματος από αντιδράσεις.	31
Σχήμα 7: Πλατφόρμα Recsip2.....	32
Σχήμα 8: Αρχιτεκτονική Recsip2.	33
Σχήμα 9: Υλοποίηση Hardware ειδικού σκοπού για στοχαστική προσομοίωση βιοχημικών μοντέλων.....	34
Σχήμα 10: Αρχιτεκτονική GPU.....	36
Σχήμα 11: Υλοποίηση αρχιτεκτονικής FRM-NRM SoC με 4 επεξεργαστές. Η διευθυνσιοδότηση φαίνεται με μώβ γραμμές, τα σήματα ελέγχου με γκρί και ο διάυλος δεδομένων με μαύρο χρώμα.	38
Σχήμα 12: Απεικόνιση μορίων προς αντίδραση.	41
Σχήμα 13: Υπολογισμός εξέλιξης των στοιχείων της βιοχημικής αντίδρασης με βάση τη στοχαστική προσομοίωση.	44
Σχήμα 14: Στιγμιότυπα χρόνου αντίδρασης.	46
Σχήμα 15: Περιγραφή της εξέλιξης μιας βιοχημικής αντίδρασης με βάση τη στοιχειομετρία.....	47
Σχήμα 16: Παραδείγματα Στοχαστικών προσομοιώσεων.	47
Σχήμα 17: Προσομοιωτής Copasi.....	51
Σχήμα 18: Παράδειγμα Στοχαστικής προσομοίωσης με το Copasi.....	52

Σχήμα 19: Απεικόνιση δικτυώματος αντιδράσεων και αλληλεξαρτήσεων από τον προσομοιωτή Ibiosim tool.	52
Σχήμα 20: Παράδειγμα προσομοίωσης με το εργαλείο Ibiosim.	53
Σχήμα 21: Ανάλυση διαγράμματος αρχιτεκτονικής FRM διαχωρισμένο σε τομείς με N=4 επεξεργαστές.....	56
Σχήμα 22: Διαδρομή δεδομένων στο FRM4X SoC για τον υπολογισμό της επόμενης αντίδρασης.....	60
Σχήμα 23: Διαδρομή δεδομένων στο FRM4X SoC για τον υπολογισμό των συγκεντρώσεων των στοιχείων αντίδρασης.....	61
Σχήμα 24: Απεικόνιση εκτέλεσης της FRM προσομοίωσης με N=1 επεξεργαστή και N=4 παράλληλους.....	62
Σχήμα 25: Διαδρομή δεδομένων στο FRM4X SoC για τον υπολογισμό των N επόμενων αντιδράσεων σε τρόπο λειτουργίας'2'.	63
Σχήμα 26: Διαδρομή δεδομένων στο FRM4X SoC στον 2 ^ο τρόπο λειτουργίας για τον υπολογισμό N διαφορετικών συγκεντρώσεων των στοιχείων των N διαφορετικών βιομοντέλων.	63
Σχήμα 27: Διάγραμμα κατανομής των εργασιών (threads) για την εκτέλεση του FRM. ...	65
Σχήμα 28: Διάγραμμα δένδροειδούς δομής των MTUS.	65
Σχήμα 29: Διάγραμμα Αρχιτεκτονικής FRM με N=4 επεξεργαστές για 1 και N παράλληλες προσομοιώσεις.....	68
Σχήμα 30: Διάγραμμα με παράλληλα νήματα (threads) στην κάθε λειτουργική υπόμοναδα για την εκτέλεση N παράλληλων FRM.....	69
Σχήμα 31: Οργάνωση μνημών (Species Table και Reaction table για μια παράλληλη προσομοίωση)	70
Σχήμα 32: Οργάνωση μνημών (Species Table και Reaction table για N παράλληλες προσομοιώσεις).....	71
Σχήμα 33: Διάγραμμα απεικόνισης υπό-μονάδας PE για μέχρι 3 ^{ου} βαθμού αντιδράσεις.	73
Σχήμα 34: Διάγραμμα απεικόνισης λειτουργίας και προγραμματισμού για τον υπολογισμό της πιθανότητας αντίδρασης.	74
Σχήμα 35: Προσέγγιση λογαρίθμου με βάση πολυωνυμική εξίσωση 2 ^{ου} βαθμού.	76
Σχήμα 36: Μονάδα LUT Xilinx.	80

Σχήμα 37: Διάγραμμα αρχιτεκτονικής FRM-NRM με 4 επεξεργαστές.	84
Σχήμα 38: Περιγραφή βασικών υπό μονάδων FRM-NRM με 4 επεξεργαστές.	85
Σχήμα 39: Οργάνωση γράφου εξαρτήσεων	88
Σχήμα 40: Περιγραφή Μνημών FIFO για κράτηση των δεδομένων.....	90
Σχήμα 41: Περιγραφή τρόπου πολλαπλών ταξινομήσεων.	91
Σχήμα 42: Περιγραφή μνημών για οργάνωση και αποθήκευση των υπολογιζόμενων δεδομένων σε δένδρα προτεραιότητας και για τους 2 τρόπους λειτουργίας.....	91
Σχήμα 43: Περιγραφή λειτουργίας St-block.....	95
Σχήμα 44 :Διάγραμμα επεξεργαστή Leon.	97
Σχήμα 45:Διάγραμμα βασικής κύριας μονάδας του Leon.....	98
Σχήμα 46: Διάγραμμα διασύνδεσης FRM4X _NRM4X με επεξεργαστή Leon.	100
Σχήμα 47: Διάγραμμα συστήματος FPGA -SysPy.....	102
Σχήμα 48: Προγραμματισμός εσωτερικού Control register.....	103
Σχήμα 49: Διαδικασία ανάθεσης παραμέτρων και σύνθεσης κυκλωμάτων ειδικού σκοπού μέσω της Python.....	104
Σχήμα 50 : Διάγραμμα απόδοσης εξαρτώμενο από τον αριθμό των PEs για μεγάλα βιοχημικά μοντέλα αντιδράσεων.	110
Σχήμα 51: Ταλαντώσεις Lotka Voltera για μελέτες ταλαντευόμενο συστημάτων με 2 στοιχεία. Η 1 ^η εικόνα περιγράφει ένα ταλαντευόμενο σύστημα για $X(0)=Y(0)=1000$. Το κάτω σύστημα είναι για μελέτη πληθυσμών 2 είδη πιθήκων που οι πληθυσμοί τους παρουσιάζουν ταλαντώσεις για $X(0)=40$ και $Y(0)=80$	117
Σχήμα 52: (α) Ταλαντώσεις Lotka-Voltera από βιβλιογραφία, (β) προσομοίωση μοντέλου από πλατφόρμα FRM4X με επεξεργαστή Leon σε τρόπο λειτουργίας '2' για 10 επαναλήψεις και χρόνο προσομοίωσης 181 δευτερόλεπτα.	118
Σχήμα 53: (α) Ταλαντώσεις Lotka Voltera από την προσομοίωση σε Matlab για τον αλγόριθμο του Gillespie με την Direct Method SSA όπου παρατηρούμαι ταλαντώσεις για αρχικές συγκεντρώσεις $X(0)=50$, $Y(0)=100$ και (β) ταλαντώσεις του ίδιου μοντέλου από το FRM4X SoC. Έγιναν και στις 2 προσομοιώσεις 10 FRM επαναλήψεις για χρόνο 30 δευτερολέπτων.	119
Σχήμα 54: Συμπεριφορά τυπικής ενζυματικής αντίδρασης.	120

Σχήμα 55: (α) Συμπεριφορά ενζυματικών αντιδράσεων αποτελέσματα από την προσομοίωση σε Matlab (Direct Method)για χρόνο αντίδρασης 40 sec ,10 επαναλήψεις , (β) αποτελέσματα προσομοίωσης από το FRM4X SoC τρόπο λειτουργίας 2 χρόνο προσομοίωσης 40 sec 10 επαναλήψεις.	121
Σχήμα 56: (α) Διασπορά των τιμών των συγκεντρώσεων από τις 10 επαναλήψεις (β) Μέσο τετραγωνικό σφάλμα για όλο το φάσμα αντίδρασης.	121
Σχήμα 57: Αποτελέσματα προσομοίωσης BIOMD 188 από (α) ibiosim tool (β) FRM4X για χρόνο αντίδρασης 3000 secs για την συμπεριφορά των στοιχείων,p53, Mdm2 για την βιοχημική ταλάντωση ARF.....	123
Σχήμα 58: (α) Διασπορά τιμών των συγκεντρώσεων από τις 8 προσομοιώσεις του BIOMD 188 από FRM4X SoC για χρόνο αντίδρασης 3,000 sec τρόπο λειτουργίας '2' (β) MSE ανάλυση για τις πρωτείνες p53, Mdm2.....	124
Σχήμα 59: (α) Αποτελέσματα προσομοίωσης BIOMD 189 από το FRM4X SoC για χρόνο αντίδρασης 2500 sec όπου φαίνεται η συμπεριφορά των στοιχείων Mdm2,p53 για το ATM μοντέλο από την προσομοίωση στο Ibiosim Tool. Ενώ στο (β) φαίνονται τα αποτελέσματα από την αντίστοιχη προσομοίωση στο FRM4X SoC σε τρόπο λειτουργίας'2'.....	125
Σχήμα 60: (α) Αποτελέσματα προσομοίωσης για το Biomodel 189 με χρόνο προσομοίωσης στα 5000 seconds για 8 FRM επαναλήψεις (β) αποτελέσματα από το NRM4X SoC σε τρόπο λειτουργίας'2' για 5000 second για 8 επαναλήψεις	125
Σχήμα 61: Στο (α) και στο (β) παρατηρούμε το Μέσο Τετραγωνικό Σφάλμα MSE για το μοντέλο 189 για τους προσομοιωτές NRM4X και FRM4X. Στα σχήματα (γ) και (δ) βλέπουμε αντίστοιχα για το FRM4X και για το NRM4X την διασπορά στις τιμές.....	126
Σχήμα 62: Αποτελέσματα από την διεθνή βιβλιογραφία από προσομοιώσεις για χρόνο αντίδρασης 30h για την συμπεριφορά των στοιχείων Mdm2,p53, για το ATM μοντέλο και το ARF αντίστοιχα.....	127
Σχήμα 63: Αποτελέσματα προσομοίωσης (a) BIOMD 188 189 από βιβλιογραφία. (b),(c) αποτελέσματα από FRM4X για χρόνο αντίδρασης 144000 second 30h για την συμπεριφορά των στοιχείων Mdm2,p53, για το ATM μοντέλο και το ARF αντίστοιχα..	127
Σχήμα 64: Παρατηρούμαι στο σχήμα (α) τις μέσες τιμές των συγκεντρώσεων των στοιχείων ενδιαφέροντος από τον προσομοιωτή Copasi. Στο (β) αντίστοιχα παρατηρούμε τις μέσες τιμές τουFRM4X SoC σε τρόπο λειτουργίας '1'για 8 επαναλήψεις του FRM αλγορίθμου ορίζοντας τον χρόνο προσομοίωσης στα 600,000 δευτερόλεπτα.	129
Σχήμα 65: (α) Μέσο συστηματικό σφάλμα για τα στοιχεία ενδιαφέροντος (παράγοντας Lamp2a,μονομερές Wild Type asyn και Dopamine Wild Type asyn) (β) Διασπορά τιμών συγκεντρώσεων από το FRM4X SoC για 8 επαναλήψεις.....	130
Σχήμα 66: (α) Αποτελέσματα προσομοίωσης από το Ibiosim για 8 διαφορετικές προσομοιώσεις για χρόνο προσομοίωσης 90,000 sec,(β) Αποτελέσματα από την προσομοίωση του FRM αλγορίθμου 8 επαναλήψεις, χρόνος προσομοίωσης 90,000.	133

Σχήμα 67: Αποτελέσματα προσομοίωσης στοιχείων ενδιαφέροντος (α) για τις προσομοιώσεις στο FRM4X και (β) για τον προσομοιωτή NRM4X σε τρόπο λειτουργίας '1' με χρόνο προσομοίωσης 90,000 second για 8 επαναλήψεις. Ακολούθως τα διαγράμματα (γ)(δ) και (ε) δείχνουν τα αποτελέσματα αντίστοιχων προσομοιώσεων για το UCHL1 ND σύστημα 133

Σχήμα 68: Αποτελέσματα προσομοίωσης στοιχείων ενδιαφέροντος (α) Μέσο συστηματικό σφάλμα για τον αλγόριθμο FRM για τις μέσες τιμές από το Copasi συγκρινόμενες με τις τιμές του FRM4X SoC. (β) Η αντίστοιχη διακύμανση τιμών από τις 8 προσομοιώσεις του FRM4X σε τρόπο λειτουργίας '1' για χρόνο προσομοίωσης 90,000 second. Ακολούθως το διάγραμμα (γ) δείχνει το MSE για τις προσομοιώσεις στο NRM4X συγκρινόμενες με τις προσομοιώσεις στο Copasi και αντίστοιχα στο (δ) φαίνεται η διακύμανση των τιμών για την NRM προσομοίωση..... 134

Σχήμα 69: Αποτελέσματα προσομοίωσης in Vitro για το Heat response of E-Coli όπου φαίνονται απότομες μεταβολές θερμοκρασίας και σταθεροποίηση..... 135

Σχήμα 70: Αποτελέσματα προσομοιώσεων HRS από iBiosim tool για χρόνο αντίδρασης 1000 δευτερολέπτων συμπεριφορά στοιχείων X1,X5,X6 καθώς και για όλα τα στοιχεία. 136

Σχήμα 71: Αποτελέσματα προσομοίωσης HRS από FRM2X SoC για χρόνο αντίδρασης 1000 second συμπεριφορά στοιχείων X1,X6,X18,X10 σε τρόπο λειτουργίας '1'. 136

Σχήμα 72: Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων από την διεθνή βιβλιογραφία για την παραγωγή των πρωτεϊνών GFP και CT..... 138

Σχήμα 73 : Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων από Ibiosim tool για χρόνο αντίδρασης 100 secs για την συμπεριφορά των στοιχείων 1x-7x. 138

Σχήμα 74: Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων FRM4X SoC για χρόνο αντίδρασης 400 min συμπεριφορά στοιχείων 2x,3x,5x..... 140

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Υπολογισμός όλων των περιπτώσεων της πιθανότητας αντίδρασης για αντιδράσεις μέχρι 3^{ου} βαθμού.

Πίνακας 2: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα PE.

Πίνακας 3: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα FRM1X .

Πίνακας 4: Στατιστικά στοιχεία για τις βασικές υπό-μονάδες των FRM4X_NRM4X SoC.

Πίνακας 5: Στατιστικά στοιχεία P.P.R προσομοίωσης και αναφορά επιδόσεων, εσωτερικών καθυστερήσεων και κατανομής χώρου για τα FRM1X-2X -4X-8XSoC.

Πίνακας 6: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα PE σχεδίαση σε Altera για 2^{ου} βαθμού αντιδράσεις.

Πίνακας 7: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα FRM1A σχεδίαση σε Altera.

Πίνακας 8: Στατιστικά στοιχεία P.P.R προσομοίωσης για επιδόσεις Mreactions/sec, εσωτερικές καθυστερήσεις και απόδοσης (MRC/sec) για τα FRM1X-1A -2A-4A SoCs σχεδίαση σε Altera

Πίνακας9: Στατιστικά στοιχεία P.P.R προσομοίωσης για επιδόσεις M-reactions/sec, καθυστερήσεις και απόδοση λειτουργίας για την FRM4X-NRM4X SoC σχεδίαση σε Xilinx Virtex 5-7 FPGAs.

Πίνακας 10: Γενικοί Παράμετροι σχεδίασης.

Πίνακας 11: Στατιστικά στοιχεία P.P.R προσομοίωσης για επιδόσεις Mreactions/sec, καθυστερήσεων και απόδοσης για την FRM1X-FRM2X-FRM4X-FRM4X_NRM4X SoC σχεδίαση σε Xilinx.

Πίνακας 12: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση με FRM σε C++ για διάφορες CPU.

Πίνακας 13: Χρόνοι υπολογισμού για όλες τις διαδικασίες του NRM σε C++ για την προσομοίωση των μοντέλων 293 και HRS σε επεξεργαστή i7 CPU της Intel.

Πίνακας 14: Χρόνοι υπολογισμού για όλες τις διαδικασίες του NRM από το FRM_NRM4X σε τρόπο λειτουργίας 1 για τον NRM αλγόριθμο για την προσομοίωση των μοντέλων 293 και HRS.

Πίνακας 15: Χρόνοι προσομοίωσης του μοντέλου 293 στο NRM4X SoC σε CPU και αντίστοιχα σε GPU.

Πίνακας 16: Σύγκριση επιδόσεων μεταξύ των 2 προσομοιωτών NRM4X και FRM4X.

Πίνακας 17: Απόδοση συστήματος FRM4X SoC+Leon για το Lotka-Volterra μοντέλο για μία end to end διαδικασία.

Πίνακας 18: Απόδοση συστήματος FRM4X SoC+Leon για το Lotka-Voltera2 μοντέλο για μία end to end διαδικασία.

Πίνακας 19: Απόδοση συστήματος FRM4X SoC+Leon για την Ενζυματική αντίδραση για μία end to end διαδικασία.

Πίνακας 20: Μέσο τετραγωνικό σφάλμα και μέσο σφάλμα για τα στοιχεία ενδιαφέροντος.

Πίνακας 21: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel 188 για μία end to end διαδικασία.

Πίνακας 22: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel 189 για μία end to end διαδικασία.

Πίνακας 23: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel 189 για μία end to end διαδικασία.

Πίνακας 24: Απόδοση συστήματος προσομοίωσης NRM4X SoC+Leon για το biomodel 189 για μία end to end διαδικασία.

Πίνακας 25: Στατιστικά στοιχεία επιδόσεων (speedup) και σύγκριση των προσομοιώσεων στο FRM4X και στο Ibiosim για τα Biomodels 188 και 189.

Πίνακας 26: Στατιστικά στοιχεία επιδόσεων (speedup) και σύγκριση με FRM4X και Ibiosim για το Biomodels 189 για χρόνο προσομοίωσης 30 ωρών.

Πίνακας 27: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel Asyn για μία end to end διαδικασία.

Πίνακας 28: Μέσο τετραγωνικό σφάλμα για τα στοιχεία ενδιαφέροντος από τους προσομοιωτές NRM4X και FRM4X για την προσομοίωση του μοντέλου UCHL1.

Πίνακας 29: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση με FRM2X και Ibiosim για το HRS biomodel.

Πίνακας 30: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση με FRM4X και Ibiosim για το Autoregulatory Gene Network biomodel

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στα πλαίσια της φοίτησης μου στο Μεταπτυχιακό Πρόγραμμα Σπουδών της Μικροηλεκτρονικής που διεξάγεται στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Με την ευκαιρία αυτή θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου κ. Ηλία Μανωλάκο, αναπληρωτή καθηγητή ΕΚΠΑ, που μου έδωσε τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέροντα τομέα της Συστημικής Βιολογίας και της Βιοπληροφορικής. Καθώς επίσης τον υποψήφιο διδάκτορα Βαγγέλη Λογαρά που η πολύτιμη έρευνα και βοήθεια που παρείχε στον τομέα της Πληροφορικής βοήθησε στην εξέλιξη του συστήματός μας. Επίσης θα ήθελα να εκφράσω τις βαθιές μου ευχαριστίες στους Δημήτρη Μανατάκη και Σίμο Χουβαρδά για την σημαντική βοήθεια τους κατά την διάρκεια ανάπτυξης και εκπόνησης της παρούσας διπλωματικής εργασίας καθώς και τον μεταπτυχιακό φοιτητή Λευτέρη Ουζούνουγλου για την πολύτιμη βοήθεια του στον τομέα της Βιοπληροφορικής. Τέλος θα ήθελα να ευχαριστήσω τη Διευθύντρια του Μεταπτυχιακού προγράμματος σπουδών της Μικροηλεκτρονικής και καθηγήτρια του τμήματος Πληροφορικής κ. Αγγελική Αραπογιάννη για τη πολύτιμη βοήθεια της κατά τη διάρκεια των σπουδών μου.

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή

Στο κεφάλαιο αυτό θα ξεκινήσουμε την ανάλυση μας με μια γενική περιγραφή για την έννοια και την εξέλιξη της συστημικής βιολογίας και της βιοπληροφορικής. Επίσης θα αναλύσουμε συνοπτικά την δομή και την τεχνολογία των FPGAs που χρησιμοποιήθηκαν. Επιπλέον θα αναφερθούμε σε παρόμοιες εργασίες από την βιβλιογραφία καθώς επίσης θα υπάρξει μια γενική εισαγωγή στη παρούσα εργασία και τη συνεισφορά αυτής γενικότερα στον τομέα της Συστημικής βιολογίας.

Η Συστημική βιολογία είναι ένας καινούργιος τομέας της βιολογίας και κατά συνέπεια και της βιοπληροφορικής που ασχολείται με την κατανόηση και ανάλυση με βάση αλγορίθμων, πολύπλοκων βιοχημικών μοντέλων για την κατανόηση ασθενειών, ιών, γονιδίων, ανάλυση πρωτεϊνών και γενικότερα αντιδράσεων του μεταβολισμού που αφορούν έμβια όντα και γενικότερα μικροοργανισμών. Επίσης μέσω της βιοπληροφορικής και της κατανόησης των αντιδράσεων του μεταβολισμού υπάρχει η δυνατότητα δημιουργίας εξελιγμένης μορφής «έξυπνων» φαρμάκων με απώτερο σκοπό τη στοχευόμενη δράση και αντιμετώπιση των ασθενειών.

Στον τομέα αυτό σημειώνετε ραγδαία επιστημονική ανάπτυξη με αποτέλεσμα την συνεχή αναγκαιότητα για ανάπτυξη καινούργιων καινοτόμων τεχνολογικών μεθόδων, για την μελέτη και ανάλυση βιοχημικών και βιολογικών δικτύων από χιλιάδες βιοχημικές αντιδράσεις. Παραδείγματα τέτοιων πολύπλοκων συστημάτων αφορούν τη μελέτη και ανάλυση μοντέλων που προσομοιώνουν το ανοσοποιητικό σύστημα έμβιων όντων[4] καθώς και την ανάλυση του γονιδιώματος για την μελέτη της ανάπτυξης των πρωτεϊνών. Όπως μπορεί να γίνει αντιληπτό η συνεχής ανάπτυξη αυτού του κλάδου προκαλεί την περαιτέρω εξέλιξη της τεχνολογίας καθώς οι απαιτήσεις σε υπολογιστική δύναμη συνεχώς αυξάνονται. Η ανάλυση τέτοιων συστημάτων απαιτεί επίσης την ανάπτυξη μιας παγκόσμιας εμβέλειας ψηφιακής βιβλιοθήκης όπου οι επιστήμονες και οι βιολόγοι να μπορούν να καταγράψουν και να κοινοποιούν τις αναλύσεις των βιοχημικών αντιδράσεων.

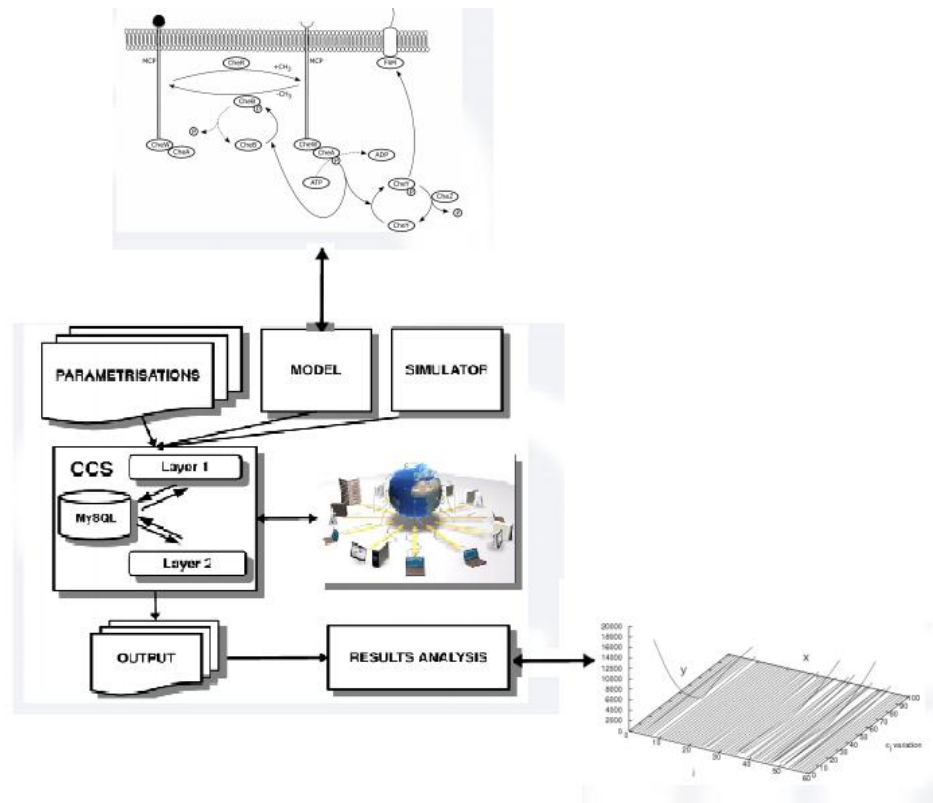
Το 2000 δημιουργήθηκε η πρώτη βάση δεδομένων των πειραματικά επιβεβαιωμένων πρωτεϊνικών αλληλεπιδράσεων, η DIP. Οι βάσεις δεδομένων, όπως η DIP, αποδείχθηκαν να είναι ουσιαστικά η «τροφή» που χρειαζόταν για να δημιουργηθεί ο τομέας της

Συστημικής Βιολογίας. Οι βιολόγοι, ακολουθώντας την ως τότε κυρίαρχη- αλλά όχι και μοναδική-προσέγγιση, την αναλυτική (analytical) ή αναγωγική (reductionist) προσέγγιση, ερευνούσαν καθεμιά από τις πρωτεϊνικές αλληλεπιδράσεις ξεχωριστά ή σε μικρές ομάδες (βιολογικές οδούς). Αντιθέτως, όταν οι μαθηματικοί, οι φυσικοί και οι πληροφορικοί κοίταξαν τις βάσεις δεδομένων που μόλις είχαν δημιουργηθεί, είδαν τις αλληλεπιδράσεις ως κομμάτια ενός παζλ, ενός συστήματος.

Με άλλα λόγια, η Βιοπληροφορική δεν τακτοποίησε απλά τα μεγάλα διαστάσεων πια βιολογικά δεδομένα, αλλά δημιούργησε χάρτες (η DIP είναι χάρτης των δικτύων αλληλεπιδράσεων (interactome), ενώ άλλες βάσεις δεδομένων είναι οι χάρτες του γονιδιώματος (genome), του πρωτεϊνιδιώματος (proteome), του μεταγραφώματος (transcriptome) κτλ.). Άρα, η μεγάλη συνεισφορά της Βιοπληροφορικής ήταν ότι έδειξε μια διαφορετική καινούργια- προσέγγιση των βιολογικών δεδομένων που υπήρχαν ως τότε.

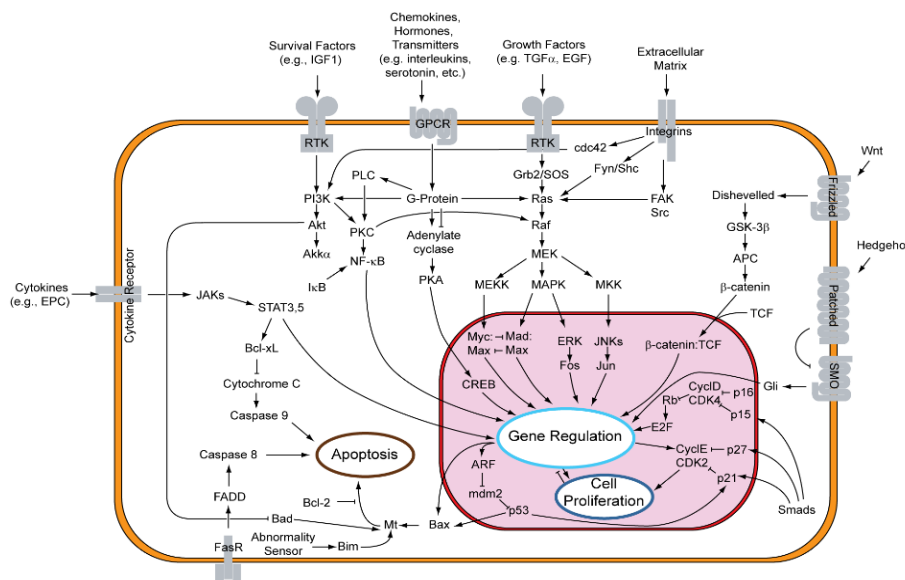
Από εκεί και πέρα, θα μπορούσαμε να πούμε ότι η Συστημική Βιολογία «καταβρόχθισε» τη Βιοπληροφορική και πήρε τη θέση της στο επίκεντρο του ενδιαφέροντος. Φυσικοί, μαθηματικοί και πληροφορικοί παρατήρησαν ότι θα μπορούσαν να χρησιμοποιήσουν τις τεχνικές που είχαν αναπτύξει για τα δικά τους δεδομένα, για να αναλύσουν τα βιολογικά συστήματα. Το επιχείρημά τους ήταν ότι μέσα από έναν επαναληπτικό κύκλο μοντελοποίησης, προσομοίωσης και θεωρίας θα βοηθούσαν στην κατανόηση των βιολογικών συστημάτων και στη στόχευση των πειραμάτων, διατυπώνοντας νέες υποθέσεις οι οποίες θα μπορούσαν να διερευνηθούν ως προς την εγκυρότητά τους στα εργαστήρια από τους βιολόγους.

Αυτό οδήγησε στη δημιουργία της SBML μια ψηφιακής βιβλιοθήκης ανάλυσης βιομοντέλων που περιγράφει κωδικοποιημένα σε xml μορφή, εξισώσεις που αφορούν την ανάλυση και περιγραφή της βιοχημείας τέτοιων βιοχημικών συστημάτων.



Σχήμα 1: Περιγραφή της ανάλυσης βιοχημικών αντιδράσεων μέσα από τη συστημική βιολογία.

Με αυτό τον τρόπο μπορεί να γίνει σε επίπεδο ενός κυττάρου η ανάλυση και η μελέτη όλων των βιοχημικών αντιδράσεων που λαμβάνουν χώρα στο κύτταρο. Το παρακάτω διάγραμμα αναπαριστά βιοχημικές αντιδράσεις στο κύτταρο οι οποίες μπορούν να περιγραφούν σε SBML.



Σχήμα 2: Μοντέλο μελέτης βιοχημικών αντιδράσεων στο κύτταρο.

Η εξέλιξη της πληροφορικής και κατά συνέπεια της βιοπληροφορικής παίζει το πλέον καθοριστικό ρόλο. Υπάρχει λοιπόν η συνεχής ανάγκη για ανάπτυξη καινούργιων αλγορίθμων που αφορούν την ανάλυση πολύπλοκων βιοχημικών συστημάτων οι οποίοι

να είναι και αποδοτικοί όσο αναφορά την επεξεργαστική τους πολυπλοκότητα για τη μοντελοποίηση βιοχημικών συστημάτων σε κάποιο Η/Υ ή κάποια επεξεργαστική μονάδα. Επίσης ένας άλλος σημαντικός παράγοντας είναι η τεχνολογία των Η/Υ και γενικότερα των επεξεργαστών που καλούνται να αντεπεξέλθουν και να αναλύσουν πολύπλοκες μαθηματικές διεργασίες με βάση τους αναπτυσσόμενους αλγόριθμους. Αυτό οδήγησε και σε καινούργιες αρχιτεκτονικές σχεδιάσεις μικρό-υπολογιστικών συστημάτων όπως νέας γενιάς GPU και CPU και κυρίως σε clusters από CPUs και GPUs καθώς και στην δημιουργία επιστημονικών Η/Υ (power PCs). Η έρευνα της τελευταίας δεκαετίας κατέστησε αναγκαία και την ανάπτυξη συστημάτων (Hardware υλοποίησής) ειδικού σκοπού για την προσομοίωση και μοντελοποίηση βιοχημικών συστημάτων.

Ενδιαφέρων επίσης παρουσιάζει και η υλοποίηση αναδομήσιμων (Reconfigurable) αρχιτεκτονικών μικρο-υπολογιστικών συστημάτων ειδικού σκοπού με την χρήση της τεχνολογίας (FPGA). Η τεχνολογία FPGA επιτρέπει την συνεχή ανάπτυξη του υλικού (hardware) ανάλογα με τις επεξεργαστικές απαιτήσεις του κάθε αλγορίθμου. Επίσης η τεχνολογία των FPGA εγγυάται και την σωστή αξιοποίηση του εκάστοτε συστήματος σε ASIC μορφή. Σημαντικός επίσης παράγων είναι η συνεχής τεχνολογική ανάπτυξη των FPGA με χαρακτηριστικό παράδειγμα την κατηγορία αναπτυξιακών συστημάτων FPGA (Virtex 7) με χωρητικότητα σε λογικά στοιχεία πλέον ανώτερη του 1Mega και με συχνότητα λειτουργίας στα 1GHz, το οποίο επιτρέπει υλοποιήσεις που επιλύουν απαιτητικής δυσκολίας προβλήματα που αφορούν τον αναπτυσσόμενο τομέα της συστημικής βιολογίας.

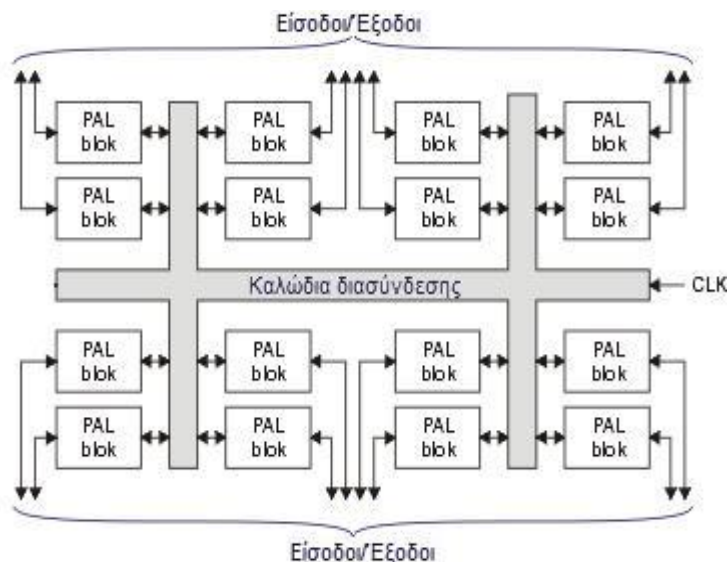
1.2 Εισαγωγή στα CPLDs και FPGAs

Σε αυτό το κεφάλαιο θα γίνει μια σύντομη περιγραφή των διατάξεων προγραμματιζόμενης λογικής (programmable logic devices) καθώς οι προδιαγραφές της εργασίας απαιτούν τη γνώση της εσωτερικής τους δομής. Στα ψηφιακά ηλεκτρονικά υπάρχουν ολοκληρωμένα κυκλώματα τα οποία έχουν σχεδιαστεί για να εκτελούν συγκεκριμένες εργασίες και άλλα τα οποία προγραμματίζονται από εμάς για διάφορες εφαρμογές, υλοποιώντας κάθε φορά το λογικό κύκλωμα που μας εξυπηρετεί. Τα κυκλώματα αυτά αποτελούνται από PLDs. Η κατάλληλη οργάνωση των διακοπών αυτών μας επιτρέπει να υλοποιούμε όποια εφαρμογή θέλουμε.

Ο προγραμματισμός των ολοκληρωμένων κυκλωμάτων όπως αναφέρθηκε παραπάνω είναι πολύ σημαντικός διότι πλέον μπορούμε να αλλάξουμε κάποια λειτουργία στο κύκλωμα μας παρεμβαίνοντας μόνο σε επίπεδο προγράμματος (software). Ένα ακόμα πλεονέκτημά τους είναι η κατασκευή πάρα πολύ μεγάλων λογικών κυκλωμάτων σε σχέση με τα τυπικά λογικά κυκλώματα. Ας δούμε τώρα ποια είδη προγραμματιζόμενων λογικών διατάξεων έχουμε. Υπάρχουν όπως αναφέραμε τα FPGAs και τα CPLDs.

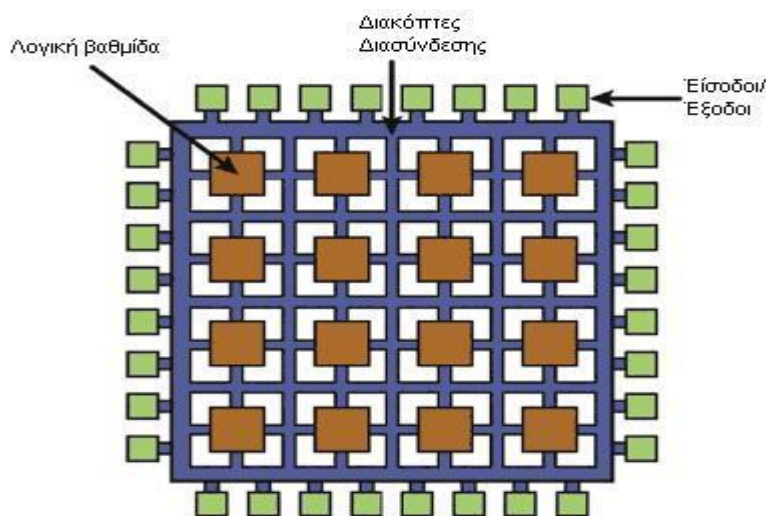
Το CPLD αποτελείται από πολλές βαθμίδες λογικών κυκλωμάτων που συνδέονται μεταξύ τους. Οι βαθμίδες αυτές ονομάζονται και διατάξεις προγραμματιζόμενης λογικής (Programmable Array Logic, PAL). Στην εικόνα 3 μπορούμε να διακρίνουμε την εσωτερική δομή ενός CPLD.

Οι εσωτερικές καλωδιώσεις περιέχουν προγραμματιζόμενους διακόπτες οι οποίοι χρησιμοποιούνται για να συνδέουν μεταξύ τους τις βαθμίδες PAL ανάλογα με το κύκλωμα που θέλουμε να υλοποιήσουμε. Ένα μειονέκτημα αυτής της μεθόδου είναι ότι κλείνοντας κάποιους διακόπτες, διακόπτουν τη λειτουργία σε κάποιους άλλους με αποτέλεσμα σε ένα τέτοιο ολοκληρωμένο κύκλωμα να μπορούν να δουλέψουν πλήρως τα 2/3 του συνολικού ολοκληρωμένου (αυτό ισχύει και στα FPGA). Μπορούμε να βρούμε CPLDs τα οποία περιέχουν έως 2210 βαθμίδες PALs ανάλογα με την εφαρμογή που θέλουμε να πραγματοποιήσουμε.



Σχήμα 3: Δομή μιας πολύπλοκης διάταξης προγραμματιζόμενης λογικής (CPLD).

Το επόμενο είδος PLD είναι το FPGA. Η ανάγκη για ακόμη μεγαλύτερο αριθμό πυλών ήταν η δημιουργία των FPGA. Αυτές οι διατάξεις υποστηρίζουν την υλοποίηση μεγάλων κυκλωμάτων. Η μεγαλύτερη διαφορά στη δομή των CPLD και FPGA είναι ότι τα FPGA δεν περιέχουν πύλες AND ή OR αλλά λογικές βαθμίδες που υλοποιούν συναρτήσεις. Γενικά η δομή ενός FPGA περιέχει τρία είδη πόρων, λογικές βαθμίδες, βαθμίδες εισόδου/εξόδου για τη σύνδεση με τους ακροδέκτες και τέλος διακόπτες και γραμμές εσωτερικής διασύνδεσης όπως βλέπουμε στο σχήμα 4. Κάθε λογική βαθμίδα αποτελείται από ένα αριθμό εισόδων και μία έξοδο. Η πιο ευρέως χρησιμοποιούμενη λογική βαθμίδα είναι ο πίνακας αναζήτησης ή αναφοράς (lookup table, LUT), ο οποίος περιέχει κυψέλες αποθήκευσης (storage cells) που χρησιμοποιούνται για την υλοποίηση μίας μικρής λογικής συνάρτησης.



Σχήμα 4: Γενική δομή μίας διάταξης FPGA.

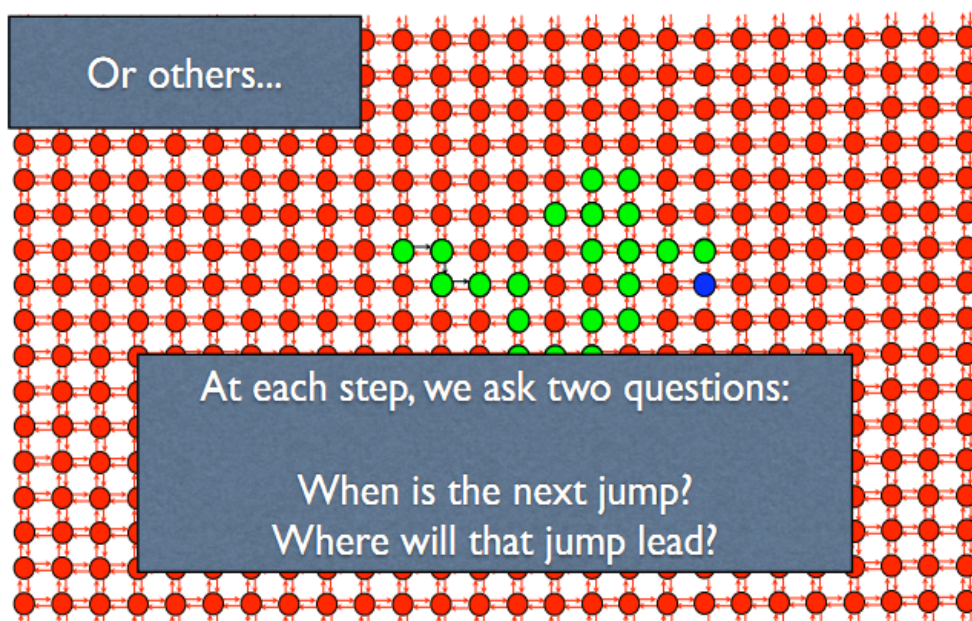
Σημαντικό μέρος στα PLDs είναι ο τρόπος προγραμματισμού τους. Μια μέθοδος που όλοι ξέρουμε είναι μέσω των βάσεων προγραμματισμού (programmer boards). Αντί να στηριζόμαστε σε μία μονάδα προγραμματισμού για να προγραμματίσουμε ένα PLD, θα ήταν χρήσιμο να μπορούμε να εκτελέσουμε τον προγραμματισμό όταν το ολοκληρωμένο κύκλωμα ευρίσκεται επάνω στην πλακέτα που θα λειτουργήσει. Για το λόγο αυτό υπάρχει η μέθοδος του προγραμματισμού μέσα στο σύστημα (In-System Programming, ISP). Τα δεδομένα για τον προγραμματισμό του, μεταδίδονται σε αυτό μέσω τη θύρας JTAG (Joint Test Action Group).

Τέλος πρέπει να γνωρίζουμε πως υπάρχει μεγάλη διαφορά αφού προγραμματίσουμε ένα FPGA ή ένα CPLD. Οι κυψέλες αποθήκευσης των πινάκων LUT των FPGAs είναι πτητικές ή μη μόνιμες (volatile), που σημαίνει ότι τα δεδομένα του προγράμματος που έχουμε στείλει από τον υπολογιστή μας θα χαθούν αφού διακόψουμε την τροφοδοσία. Γι' αυτό το λόγο τα FPGA συνοδεύονται στην πλακέτα λειτουργίας τους από ένα ολοκληρωμένο κύκλωμα μνήμης FLASH, η οποία διατηρεί το πρόγραμμα μας και ύστερα κατά την εκκίνηση του συστήματος το φορτώνει στις κυψέλες των LUT, η διαδικασία αυτή ονομάζεται φόρτωση κατά την εκκίνηση (bootload). Αντιθέτως το πρόγραμμα το οποίο θα στείλουμε σε ένα CPLD δεν χάνεται ακόμη κι εάν δεν υπάρχει τροφοδοσία, αυτό το είδος προγραμματισμού ονομάζεται μόνιμος ή αμετάβλητος προγραμματισμός (non-volatile programming).

1.3 Στοχαστικές προσομοιώσεις βιοχημικών αντιδράσεων

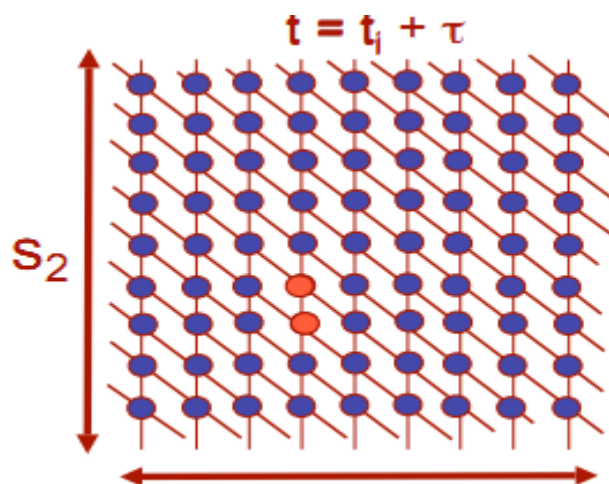
Η ανάγκη για την ύπαρξη γρήγορων μικρής πολυπλοκότητας αλγόριθμων που αναλύουν ένα βιοχημικό φαινόμενο οδήγησε στην υλοποίηση των στοχαστικών προσομοιώσεων και στην εύρεση στοχαστικών αλγορίθμων. Η πλήρης περιγραφή και ανάλυση ενός βιοχημικού φαινομένου απαιτεί την επίλυση πολύπλοκων διαφορικών εξισώσεων που όμως περιγράφουν λεπτομερώς την κατάσταση και εξέλιξη ενός βιοχημικού συστήματος με κόστος όμως την τεράστια επεξεργαστική ισχύς λόγω της πολυπλοκότητας τέτοιων συστημάτων, καθώς επίσης και ένα τεράστιο κόστος σε επεξεργαστικό χρόνο για την διεξαγωγή κάποιο χρήσιμων αποτελεσμάτων ο οποίος τείνει στο να γίνει απαγορευτικός όσο η πολυπλοκότητα αυξάνει. Για να αντιμετωπιστούν όλα τα προηγούμενα προβλήματα κρίθηκε αναγκαία η χρήση στοχαστικών προσομοιώσεων βασιζόμενα σε Μαρκοβιανά μοντέλα. Το παρακάτω διάγραμμα εν δεικνύει την λειτουργία μιας Μαρκοβιανής μεθόδου.

A Jump-Markov description of chemical kinetics



Σχήμα 5: Περιγραφή Μαρκοβιανής Μεθόδου για προσομοίωση βιοχημικών αντιδράσεων.

Όπως περιγράφει αναλυτικά το προηγούμενο διάγραμμα σε κάθε δίκτυο βιοχημικών αντιδράσεων την κάθε χρονική στιγμή επιζητούμε να μάθουμε το ποια θα είναι η επόμενη αντίδραση που θα πραγματοποιηθεί από το δίκτυο των αντιδράσεων και το πότε αυτό θα γίνει. Σε αυτά τα ερωτήματα απαντά γρήγορα και αποτελεσματικά η στοχαστική προσομοίωση.



Σχήμα 6: Στιγμιότυπο στοχαστικής προσομοίωσης ενός δικτυώματος από αντιδράσεις.

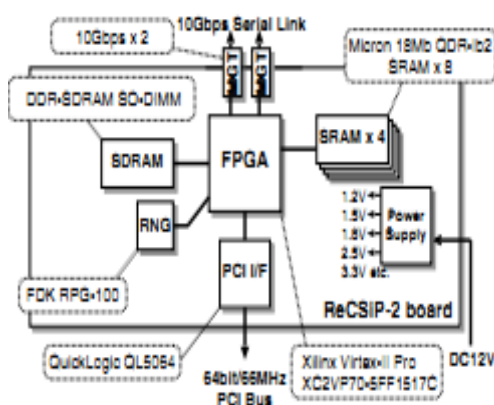
Οι Μαρκοβιανές μέθοδοι βασίζονται σε ψευδο-τυχαίες διαδικασίες. Σε κάθε χρονική στιγμή διαλέγονται κάποιοι τυχαίοι αριθμοί από μια γεννήτρια παραγωγής ψευδο- τυχαίων αριθμών. Επίσης σε κάθε χρονική στιγμή υπολογίζεται και η πιθανότητα αντίδρασης δύο ή παραπάνω ουσιών που σχετίζεται άμεσα από τις συγκεντρώσεις των στοιχείων, την εκάστοτε χρονική στιγμή και την τάση που έχει το σύστημα να αντιδράσει (reaction rate = k). Ο τυχαίος αριθμός που έχει επιλεγεί και η πιθανότητα που υπάρχει για να γίνει η κάθε αντίδραση του δικτυώματος δημιουργεί και το στιγμιότυπο του χρόνου αντίδρασης που είναι χαρακτηριστικό της αντίδρασης που έγινε. Από αυτούς τους χρόνους αντίδρασης που είναι ενδεικτικοί για την κάθε αντίδραση διαλέγετε ο μικρότερος που αντιστοιχεί στην αντίδραση που θα πραγματοποιηθεί και έτσι στο προϋπάρχον στιγμιότυπο χρόνου αντίδρασης προστίθεται ο νέος ελάχιστος χρόνος $t_{new} = t_{old} + \tau_{min}$.

Η αντίδραση η οποία θα πραγματοποιηθεί οδηγεί σε νέες συγκεντρώσεις που έχουν σχέση με την στοιχειομετρία (αριθμός σθένους μορίων με βάση την κατανομή των ηλεκτρονίων σε τροχιακά) οπότε έπειτα από κάθε αντίδραση θα έχω την κατανάλωση των αντιδρώντων για τη δημιουργία των προϊόντων με αποτέλεσμα την εξέλιξη του βιοχημικού συστήματος.

1.4 Ανάγκη για υλοποιήσεις σε επίπεδο υλικού και η σύγκριση με τις αντίστοιχες εφαρμογές σε επίπεδο λογισμικού.

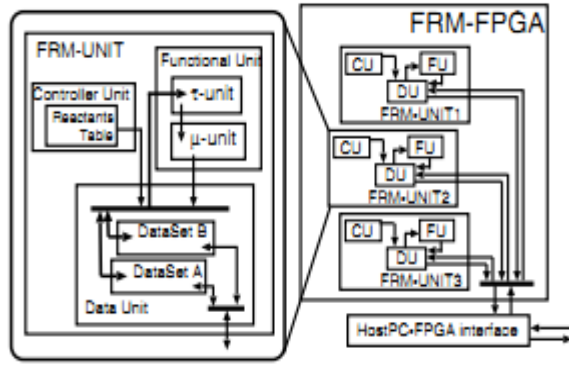
Μια από τις πιο γνωστές σχεδιάσεις προσομοιωτή στοχαστικής προσομοίωσης σε επίπεδο υλικού Hardware είναι το ReCSIP2 [5] το οποίο σχεδιάστηκε σε Xilinx στο Virtex II FPGA και υλοποιεί τον FRM (First Reaction Method) [1] αλγόριθμο στοχαστικής προσομοίωσης με μέγιστη συχνότητα λειτουργίας 107 Mhz. Η πλατφόρμα συγκρίθηκε με υλοποίηση του FRM αλγορίθμου σε C στον επεξεργαστή Xeon 2.80 GHz και βρέθηκε ότι η συγκεκριμένη υλοποίηση σε hardware είναι κατά 80 φορές πιο γρήγορη. Με βάση αυτή την υλοποίηση από τη συγκεκριμένη ερευνητική ομάδα άνοιξε ο δρόμος για περαιτέρω υλοποιήσεις σε FPGA.

Το συγκεκριμένο σύστημα επικοινωνεί μέσω μιας θύρας PCI με υπολογιστή και σε αυτό μελετήθηκε η δυνατότητα να προσομοιωθούν βιοχημικές αντιδράσεις μέχρι και 1024 αντιδράσεων. Στο παρακάτω διαγράμμα (σχήμα 7) φαίνεται το σύστημα και η αρχιτεκτονική της συγκεκριμένης υλοποίησης



Σχήμα 7: Πλατφόρμα ReCSIP2.

Στο σχήμα 8 φαίνεται η αρχιτεκτονική του συστήματος FRM. Παρακάτω μπορούμε να παρατηρήσουμε τις βασικές υπό μονάδες της σχεδίασης, όπως το τ-unit για τον υπολογισμό του χρόνου πραγματοποίησης της αντίδρασης με βάση τη θεωρία της στοχαστικής προσομοίωσης, το μ-unit για τον υπολογισμό της στοιχειομετρίας, τις μνήμες δεδομένων (Reactants Table) καθώς και μια σχεδίαση αρχιτεκτονικής με πολλές τέτοιες μονάδες εν παραλλήλω.



Σχήμα 8: Αρχιτεκτονική Recsip2.

Μία άλλη ενδιαφέρουσα υλοποίηση σε Hardware είναι η υλοποίηση ενός επιταχυντή στοχαστικής προσομοίωσης [7] που στηρίζεται στη δημιουργία hardware ειδικού σκοπού το οποίο μπορεί να γίνει αναδομήσιμο (reconfigurable) για το κάθε βιοχημικό μοντέλο. Η όλη σχεδίαση στηρίζεται στην εξίσωση Bernoulli (εξίσωση 1).

$$S_1 + S_2 \xrightarrow{k_i} S_3 \quad |\Delta t = \text{Bernoulli} (k * S_1 * S_2 * \Delta t)$$

$$0 < P[S_1 + S_2 \xrightarrow{k_i} S_3] / \Delta t = (k * S_1 * S_2 * \Delta t) \ll 1 \quad (1)$$

Συνοπτικά αυτό το οποίο υποστηρίζει αυτή η θεωρία είναι ότι για να γίνει μία αντίδραση από N αντιδράσεις θα πρέπει αυτή η αντίδραση να έχει την μεγαλύτερη πιθανότητα για να γίνει το οποίο εξαρτάται από πολλούς παράγοντες οι οποίοι συνεχώς ελέγχονται. Για μια αντίδραση της μορφής $S_1 + S_2 + \dots + S_N \xrightarrow{k} S_{N+1}$, η πιθανότητα για να πραγματοποιηθεί είναι η ακόλουθη :

$$\text{Fires} = (X_0 < k_o) \ \&\& \ ((\bar{X}_1 \ \& \ M_1) < S_1) \ \&\& \ ((\bar{X}_2 \ \& \ M_2) < S_2) \ \&\& \ \dots \ \&\& \ ((\bar{X}_N \ \& \ M_N) < S_N) \quad (2)$$

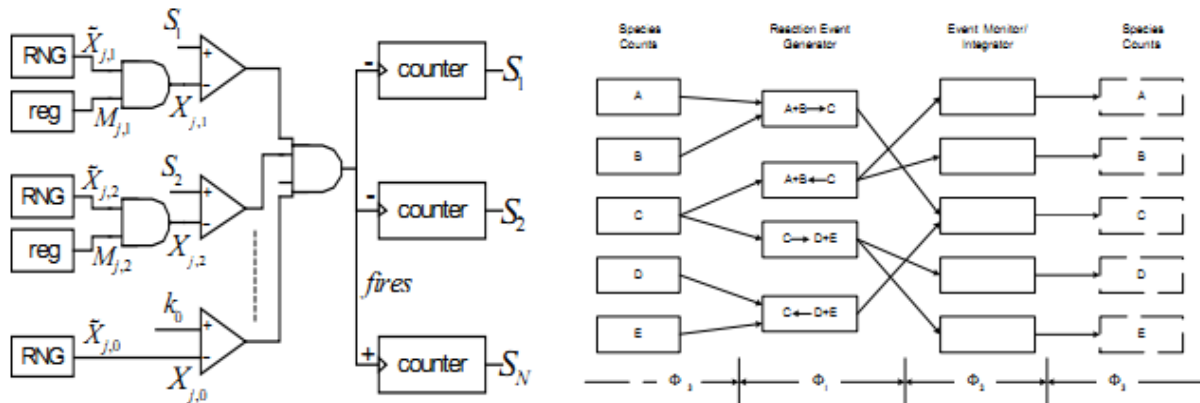
Όπου για $S_i < M_i$ όπου S_i είναι οι συγκεντρώσεις, M_i είναι ένα ανώτερο όριο, k_i είναι το reaction rate και X_i είναι ένας τυχαίος αριθμός οπότε η ολική πιθανότητα για την κάθε αντίδραση δίνεται ως εξής :

$$P[\text{fires}] = (k_o / 2^{32}) * (S_1 / (M_1 + 1)) * (S_2 / (M_2 + 1)) \quad (3)$$

Το οποίο σχετίζεται με την εξίσωση Bernoulli για ένα προκαθορισμένο χρόνο Δt με το ακόλουθο θεώρημα.

$$P[\text{fires}] = P[\text{reaction}] = \text{reaction_rate} * \text{timestep} * \text{count1} * \text{count2} * \dots * \text{countN} \quad (4)$$

Τα παρακάτω σχήματα δείχνουν τις υλοποιήσεις τις μεθόδου αυτής. Αν και σαν ιδέα η παραπάνω σχεδίαση είναι πολύ ενδιαφέρουσα τα μειονεκτήματά της είναι πολλά κυρίως όταν μιλάμε για χιλιάδες αντιδράσεις σε μεγάλα βιοχημικά μοντέλα. Επίσης πρέπει σε κάθε βήμα Δt να ελέγχονται πάντα κάποιες συγκεκριμένες μεταβλητές για να μην ξεπεράσουν κάποια ανώτατα όρια τα οποία θα οδηγήσουν το σύστημα εξισώσεων εκτός ελέγχου. Επιπλέον η όλη διαδικασία κ λόγω αυτής της συνεχούς ανάδρασης για τον έλεγχο των παραμέτρων, καθιστά την όλη υλοποίηση χρονικά αργή μειώνοντας την απόδοση.



Σχήμα 9: Υλοποίηση Hardware ειδικού σκοπού για στοχαστική προσομοίωση βιοχημικών μοντέλων με τη μέθοδο Bernoulli.

Μία άλλη ενδιαφέρουσα εφαρμογή σε επίπεδο υλικού είναι η υλοποίηση του Next Reaction Method (NRM αλγόριθμου) [2] ο οποίος θα περιγραφεί αναλυτικά στις παρακάτω παραγράφους. Η υλοποίηση του NRM σε FPGA έγινε από την ερευνητική ομάδα του Recsir που αναφέραμε προηγουμένως. Ουσιαστικά σχεδιάστηκε ένα απλό μοντέλο του NRM για να γίνει η σύγκριση με τη προηγούμενη υλοποίηση του FRM με ένα δοκιμαστικό μη πραγματικό μοντέλο 1000 αντιδράσεων 2^{ou} βαθμού δεδομένης πολυπλοκότητας. Η έρευνα που έγινε υποστηρίζει ότι ο NRM αλγόριθμος λόγω του γράφου εξαρτήσεων επιταχύνει την διαδικασία και αυξάνει την απόδοση σε σχέση με την υλοποίηση σε FRM. Επίσης η ίδια ερευνητική ομάδα πρότεινε πως για μοντέλα πολλών αντιδράσεων θα πρέπει να χρησιμοποιείτε μόνο ο NRM. Όμως η δική μας έρευνα απέδειξε ότι ο FRM λόγω της εύκολης παραλληλοποίησης του είναι ο πλέον κατάλληλος για μεγάλα συστήματα. Επίσης θα αποδείξουμε παρακάτω πως η παραλληλοποίηση που καταφέραμε με τον αλγόριθμο NRM ενώ αυξάνει την απόδοση, απαιτεί μεγάλες ανάγκες σε προγραμματιζόμενες πύλες και γενικά σε υλικό (Hardware) το οποίο δεν καθιστά την υλοποίηση τόσο εύχρηστη για FPGA εφαρμογές.

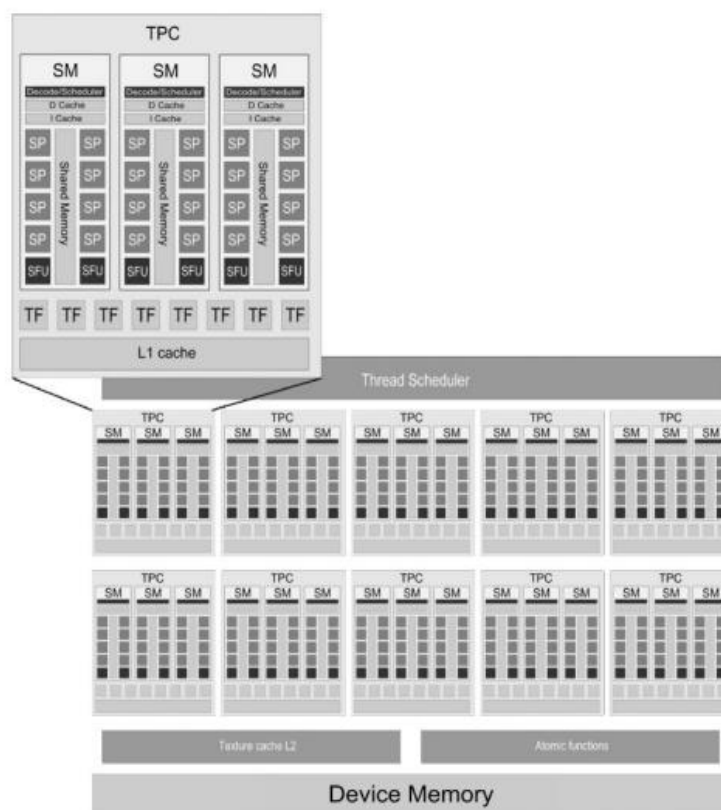
Μία άλλη ενδιαφέρουσα προσέγγιση είναι η υλοποίηση στοχαστικής προσομοίωσης σε GPU [8] με την γλώσσα προγραμματισμού της CUDA που είναι βασισμένη στην C αλλά εμπλουτισμένη με ειδικές συναρτήσεις για την Ψηφιακή επεξεργασία εικόνας. Η ιδέα

έγκειται στην ανάγκη για συνεχώς αυξανόμενη επεξεργαστική ισχύς κυρίως λόγω των συνεχών μαθηματικών υπολογισμών κινητής υποδιαστολής (floating point) που υπάρχουν στην υλοποίηση των στοχαστικών αλγορίθμων. Η αρχιτεκτονική της GPU μπορεί να θεωρηθεί κατάλληλη για τέτοιες εφαρμογές.

Οι GPUs αποτελούνται από πολλούς ομαδοποιημένους επεξεργαστές όπου ο κάθε επεξεργαστής αποτελείται από πολλές μονάδες για Αριθμητικές και Λογικές πράξεις (ALUs), όπου το Αριθμητικό κομμάτι είναι βασισμένο σε υπολογισμούς κινητής υποδιαστολής. Όλοι οι επεξεργαστές μοιράζονται την ίδια μνήμη το οποίο κρίνεται αναγκαίο για εφαρμογές ψηφιακής επεξεργασίας εικόνας.

Έχουν γίνει αρκετές προσπάθειες για στοχαστικές προσομοιώσεις βιολογικών μοντέλων σε GPUs. Χαρακτηριστικό παράδειγμα είναι μία ερευνητική ομάδα του Πανεπιστημίου της Πίζας στην Ιταλία όπου υλοποίησαν την μέθοδο FRM σε GPU. Η συγκεκριμένη ομάδα προσομείωσε ένα μοντέλο για την μετάλλαξη του βακτηριδίου E-Coli όταν αυτό υπόκειται σε υψηλές θερμοκρασίες (Heat Response of E-Coli) με 61 αντιδράσεις των 28 στοιχείων. Οι εφαρμογές έδειξαν μία επιτάχυνση της τάξης του 2 σε σχέση με μία παρόμοια προσομοίωση σε CPU για το ίδιο πρόβλημα. Η έρευνα συνεχίστηκε περαιτέρω προσπαθώντας να προσομοιωθούν μοντέλα με πολύ ανώτερο αριθμό αντιδράσεων. Εκεί η όλη διαδικασία δεν κατόρθωσε να ανταποκριθεί όσο γρήγορα αναμενόταν και αυτό οφείλεται στην αρχιτεκτονική της GPU λόγω της μικρής διαμοιραζόμενης μνήμης που διαθέτει για όλο το σύστημα των επεξεργαστών καθιστώντας την ανάλυση μεγάλων συστημάτων σε GPU μη αποτελεσματική.

Μία άλλη εφαρμογή σε GPU είναι το STOCHSIM [9] όπου είναι μία υλοποίηση σε C++ για το εργαλείο προσομοίωσης του Biology Toolbox της MATLAB το οποίο χρησιμοποιεί τις κάρτες γραφικών της NVIDIA για να επιταχύνει σε MATLAB τις διαδικασίες στοχαστικής προσομοίωσης. Φαίνεται ότι το σύστημα αυτό είναι 85 φορές πιο γρήγορο σε σχέση με μία παρόμοια υλοποίηση σε CPU.



Σχήμα 10: Αρχιτεκτονική GPU.

1.5 Κίνητρα για τη δική μας εργασία

Με βάση όλη τη προηγούμενη ανάλυση και την μέχρι τώρα πορεία των ερευνών στον τομέα αυτό κρίθηκε η ανάγκη περαιτέρω διερεύνησης των αλγορίθμων για στοχαστική προσομοίωση όσων αναφορά προσομοιώσεις με δίκτυα χιλιάδων αντιδράσεων, με γνώμονα την αύξηση την απόδοσης προσομοίωσης πολύπλοκων συστημάτων. Για αυτό το λόγο και μελετώντας την μέχρι τώρα πορεία των ερευνών δημιουργήσαμε μια αρχιτεκτονική ειδικού σκοπού [10], η οποία θα διευκολύνει τους επιστήμονες να μελετήσουν πολύπλοκα βιοχημικά φαινόμενα με βάση την στοχαστική προσομοίωση, αυξάνοντας την απόδοση σε σχέση με τα εκατομμύρια των αντιδράσεων ανά δευτερόλεπτο που μέχρι τώρα θεωρούνταν χρονοβόρο και απαγορευτικό.

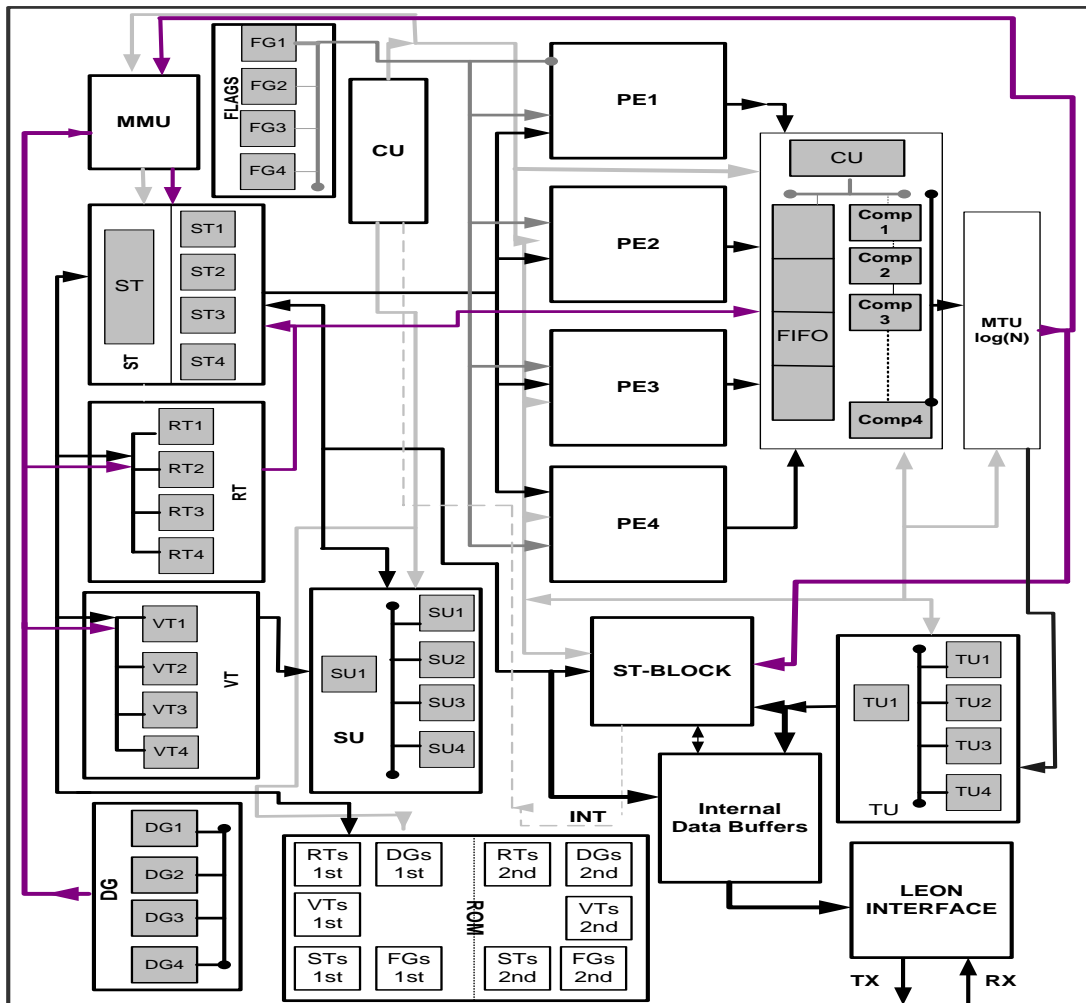
Για να γίνει εφικτό αυτό παραλληλοποιήσαμε σε επίπεδο υλικού Hardware την διαδικασία προσομοίωσης με τον FRM αλγόριθμο δημιουργώντας 4 αυτόνομες σχεδιάσεις με 1 επεξεργαστή PE (Processing Element) που υπολογίζει το χρόνο της επόμενης αντίδρασης, με 2,4 PEs αλλά και με 8 PEs αυξάνοντας θεαματικά την απόδοση των αντιδράσεων συγκρινόμενοι με παρόμοιες εργασίες και από τη διεθνής βιβλιογραφία και χρησιμοποιώντας αρεστή τεχνολογία FPGA όπως αυτή του Virtex 5 ml509. Επίσης αυτοματοποιήσαμε όλες σχεδόν τις διαδικασίες με εισαγωγή του επεξεργαστή Leon και του

προγράμματος SysPy [3] κάνοντας την λειτουργία της πλατφόρμας και του προγραμματισμού των SoCs χρηστική προς τους βιολόγους. Ακόμα κατορθώσαμε και ενσωματώσαμε και του δύο αλγορίθμους FRM και NRM στο ίδιο σύστημα SoC + Leon καθώς επίσης παραλληλοποιήσαμε τον αλγόριθμο NRM διαμοιράζοντας τον γράφο εξαρτήσεων και επιταχύνοντας την διαδικασία. Επιπλέον κατορθώσαμε μέσω της παραλληλοποίησης να δημιουργήσουμε μια ευέλικτη αρχιτεκτονική η οποία προγραμματιζόμενη μπορεί να εκτελέσει μία προσομοίωση αλλά και N διαμοιρασμένη στους N (όπου $N=2,4,8$) παράλληλους επεξεργαστές ενός μεγάλου δικτύου ή ακόμα να εκτελέσει N παράλληλες διαφορετικές προσομοιώσεις ταυτόχρονα. Με τις N παράλληλες διαφορετικές προσομοιώσεις θα μπορούν να διεξαχθούν εύκολα και γρήγορα στατιστικά συμπεράσματα και να διερευνηθούν διάφορες παραλλαγές του ίδιου βιοχημικού συστήματος. Αυτή η διαδικασία όπως θα δούμε και παρακάτω μπορεί να διεξαχθεί και με τους 2 αλγορίθμους FRM και NRM. Επίσης αναπτύχθηκαν υπό τρόποι λειτουργίας όπου να επιτρέπεται η παραλληλοποίηση της προσομοίωσης σε διαφορετικά βιοχημικά μοντέλα όταν ο αριθμός των επεξεργαστών (PEs) είναι μεγαλύτερος από τον αριθμό των μοντέλων προς προσομοίωση. Σχετικά με την πολυπλοκότητα των αντιδράσεων του δικτυώματος κατορθώσαμε και παραλληλοποιήσαμε την τάξη της αντίδρασης και στους 2 αλγορίθμους (αριθμός αντιδρώντων) και πλέον μπορούμε να προσομοιώσουμε και συστήματα ανωτέρου βαθμού μεγαλύτερα του 2 δηλαδή για παραπάνω από 3 αντιδρώντα. Τέλος να σημειωθεί πως αυτή η πλατφόρμα θα είναι χρήσιμη και για μελλοντικές χρήσεις της Συστημικής βιολογίας καθώς προβλέπετε ότι θα υπάρχει η αναγκαιότητα για μελέτη πολύ μεγάλων δικτύων από βιοχημικές αντιδράσεις γρήγορα και αποδοτικά το οποίο δεν ήταν εφικτό μέχρι σήμερα. Μέσο του ερευνητικού συστήματος SysPy που χρησιμοποιήθηκε στην παρούσα εργασία και των σχεδιάσεων των SoCs σε FPGA θα μπορεί εύκολα να δομείται κάθε φορά καλύτερο hardware (reconfigurable hardware) για την περαιτέρω βελτιστοποίηση του προσομοιωτή εξαρτωμένου των απαιτήσεων του βιοχημικού μοντέλου προς ανάλυση.

Στη παρακάτω ενότητα δίνεται μια συνοπτική περιγραφή και εισαγωγή της αρχιτεκτονικής του συστήματος που υλοποιεί τους 2 SSA αλγορίθμους (FRM-NRM). Επιπλέον η μονάδα αυτή εμπεριέχει υπομονάδα για την στατιστική επεξεργασία των συγκεντρώσεων από τις διάφορες επαναλήψεις της στοχαστικής προσομοίωσης.

1.6 Γενική περιγραφή της αρχιτεκτονικής των SoCs FRM-NRM + LEON σε Xilinx και σε ALTERA

Το παρακάτω διάγραμμα δείχνει την αρχιτεκτονική του συστήματος FRM-NRM4X δηλαδή τη σχεδίαση του συστήματος που προσομοιώνει τους αλγορίθμους First Reaction Method και Next Reaction Method με $N=4$ επεξεργαστές για υλοποιήσεις σε Xilinx αλλά και σε Altera.



Σχήμα 11: Υλοποίηση αρχιτεκτονικής FRM-NRM SoC με 4 επεξεργαστές. Η διευθυνσιοδότηση φαίνεται με μώβ γραμμές, τα σήματα ελέγχου με γκρι και ο διάυλος δεδομένων με μαύρο χρώμα.

Παρατηρούμε στο παραπάνω διάγραμμα τους 4 επεξεργαστές PEs που υπολογίζουν το χρόνο προσομοίωσης. Επίσης χρησιμοποιείτε ένα σύστημα από Block Rams όπου καταχωρούνται όλες οι αναγκαίες πληροφορίες για την ανάλυση ενός βιοχημικού μοντέλου αντιδράσεων και για τους δύο τρόπους λειτουργίας καθώς και για τους υπό τρόπους. Πιο συγκεκριμένα δομείται ο VHDL κώδικας για την αρχικοποίηση των μνημών από το SysPy με βάση το εκάστοτε SBML μοντέλο. Οι Block Rams είναι οι ST, RT VT, DG, και τα flags Table FG. Αναλυτικά η ST (species Table) είναι η μνήμη που περιέχει τις συγκεντρώσεις των στοιχείων και η RT (Reactance Table) περιέχει τη μορφή των αντιδράσεων.

Ουσιαστικά όπως θα αναλυθεί και στο κεφάλαιο 3 η μνήμη RT περιέχει δείκτες (pointers) που δείχνουν τις αντιδράσεις δείχνοντας τις συγκεντρώσεις στην ST (species Table). Συνεχίζοντας η μνήμη VT δείχνει την στοιχειομετρία της κάθε αντίδρασης δηλαδή πως θα κυμανθούν οι συγκεντρώσεις για την κάθε αντίδραση. Το DG (dependency graph memory) είναι ο γράφος εξαρτήσεων και χρησιμοποιείται όπως θα δούμε στον NRM αλγόριθμο. Τα Flag Tables FG περιέχουν συγκεκριμένες πληροφορίες στις υπολογιστικές μονάδες PEs που δείχνουν τον βαθμό της κάθε αντίδρασης πχ αν είναι 1ου, 2ου ,3ου καθώς και το τι στοιχεία αντιδρούν για να υπολογιστεί ανάλογα η εύρεση της πιθανότητας αντίδρασης. Η διευθυνσιοδότηση όλων των μνημών ελέγχεται από την MMU (Memory Management Unit). Επίσης διακρίνονται οι μονάδες SU (Species Update) για τον υπολογισμό των συγκεντρώσεων μετά την πραγματοποίηση μιας συγκεκριμένης αντίδρασης και ακολούθως οι μονάδες PE(Processing Element), υπολογίζουν το χρόνο της αντίδρασης, που εξαρτάται από τη πιθανότητα η συγκεκριμένη αντίδραση να πραγματοποιηθεί.

Ο χρόνος από όλα τα N PEs δηλαδή οι N χρόνοι αντίδρασης συλλέγονται στο δικτύωμα των βαθμίδων MTUs (Minimum Time Units) από τα οποία από μια δενδροειδής δομή υπολογίζεται ο ελάχιστος χρόνος προσομοίωσης. Τα PQ (Priority Queues) είναι δένδρα κατηγοριοποίησης που χρησιμοποιούνται για τον NRM αλγόριθμο για την εύρεση του μικρότερου χρόνου. Επίσης έχουν κατασκευαστεί για την επιτάχυνση του NRM αλγορίθμου και πολλαπλοί παράλληλοι ταξινομητές. Η μονάδα TU (Time Update Unit) χρησιμοποιείται για την αναβάθμιση του χρόνου, δηλαδή ο παλιός ο χρόνος προστίθεται στον καινούργιο $time = old + \tau(\text{minimum})$ δημιουργώντας ένα στιγμιότυπο της εξέλιξης της βιοχημικής αντίδρασης.

Επίσης όπως μπορούμε να διακρίνουμε έχουν χρησιμοποιηθεί διάφορα συστήματα επικοινωνίας τόσο με το SysPy για το Leon αλλά και DDR2 controllers που στέλνουν δεδομένα στην εσωτερική DDR2/3 του FPGA όπως επίσης σύστημα επικοινωνίας των SoC μέσω μονάδας UART. Τέλος η καρδιά όλου του συστήματος είναι η Control Unit η οποία ελέγχει την κάθε υπό-μονάδα και την όλη διαδικασία. Επιπλέον εσωτερικά βρίσκεται και ο καταχωρητής προγραμματισμού ο οποίος όπως θα δούμε χρησιμοποιείται στον προγραμματισμό όλου του συστήματος.

1.7 Τι θα δούμε στην παρούσα εργασία

Στα παρακάτω κεφάλαια θα γίνει αρχικά μια γενική ανασκόπηση και ανάλυση της φυσικής βιοχημείας πίσω από τη θεωρία της Στοχαστικής Προσομοίωσης και θα αναλυθούν οι αλγόριθμοι του Gillespie (Direct Method)[1] και η μετάβαση από τον DM στον FRM, καθώς και η εξέλιξη που έφερε η μελέτη των Gibson and Brucks για την δημιουργία του NRM αλγορίθμου. Επίσης θα υπάρξει μία εκτενής περιγραφή της αρχιτεκτονικής των FRM-NRM σχεδιάσεων καθώς και ανάλυση όλου του συστήματος SysPy και FRM4X_NRM4X SoC με τον επεξεργαστή Leon. Για την κάθε σχεδίαση θα ακολουθήσει περιγραφή και ανάλυση της απόδοσης (throughput), του χρόνου καθυστέρησης (latency), της κατανομής σε πύλες των σχεδιάσεων στο FPGA καθώς και της συχνότητας λειτουργίας όπου για όλα τα συστήματα σχεδιάσεων θα παρατεθούν αναλυτικοί πίνακες. Επίσης θα υπάρχουν συγκρίσεις με υλοποιήσεις των αλγορίθμων σε C++ για να επιδείξουμε την επιτάχυνση του συστήματος μας σε σχέση με Software υλοποιήσεις. Στα τελευταία κεφάλαια θα δείξουμε προσομοιώσεις με πραγματικά μοντέλα και θα γίνουν συγκρίσεις για την ορθότητα των αποτελεσμάτων με γνωστά χρησιμοποιούμενα εργαλεία προσομοίωσης. Τέλος θα επισημανθούν τρόποι βελτίωσης και μελλοντικές προοπτικές εξέλιξης.

ΚΕΦΑΛΑΙΟ 2

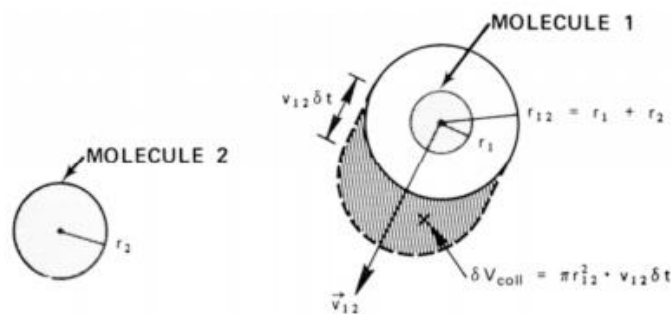
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΣΤΟΧΑΣΤΙΚΗ ΠΡΟΣΟΜΟΙΩΣΗ

2.1 Εισαγωγή στη φυσική-βιοχημεία

Ας υποθέσουμε ότι ένα βιοχημικό σύστημα αποτελείται από N ουσίες $\{S_1, \dots, S_n\}$ και από m αντιδράσεις $\{R_1, \dots, R_m\}$. Επίσης ας υποθέσουμε ότι γίνεται η υπόθεση ότι όλα τα στοιχεία της αντίδρασης βρίσκονται διάσπαρτα σε ένα συγκεκριμένο όγκο έστω Ω μέσα στο κύτταρο. Αυτές οι απλοποιήσεις μας βοηθούν στο να αναλύσουμε τα συστήματα εξισώσεων. Θέτουμε $X_i(t)$ τον αριθμό των μορίων του στοιχείου S_i την χρονική στιγμή t_i οπότε θα έχουμε ένα σύστημα της μορφής $\mathbf{X}(t) = (X_1(t), X_2(t), \dots, X_n(t))$ και αρχικές συνθήκες για το κάθε στοιχείο $\mathbf{X}_0(t) = \mathbf{x}_0$ για έναν αρχικό χρόνο $t=t_0$. Εάν στο σύστημα μας πραγματοποιηθεί μία αντίδραση έστω η R_μ τότε το υπάρχον σύστημα μας X θα ανανεώσει τα στοιχεία της συγκεκριμένης αντίδρασης κατά ένα παράγοντα ν_μ οπότε $X(t+\tau) = (X_1, \dots, X_\mu + \nu_\mu, \dots, X_n)$, όπου ο αριθμός ν_μ υποδιλώνει την στοιχειομετρία της εκάστως αντίδρασης.

Επίσης κάθε αντίδραση R_μ σχετίζεται άμεσα με έναν αριθμό c_μ που υποδιλώνει την πιθανότητα που έχουν τα αντιδρώντα να αντιδράσουν. Με βάση αυτόν τον αριθμό και τις συγκεντρώσεις υπολογίζεται η πιθανότητα που υπάρχει για να πραγματοποιηθεί κάποια αντίδραση από το σύνολο των αντιδράσεων.

Σε μοριακό επίπεδο η πιθανότητα δύο μορίων να αντιδράσουν για ένα χρονικό διάστημα Δt εκφράζεται από τον αριθμό c_μ . Αναλύοντας την φυσική θεωρούμε τα μόρια στην απλούστατη περίπτωση σαν σφαίρες όπως φαίνεται στο σχήμα 12 τα οποία συνεχώς κινούνται.



Σχήμα 12: Απεικόνιση μορίων προς αντίδραση.

Η σχετική ταχύτητα που θα βλέπει ένα από τα κινούμενα μόρια το οποίο θα προσεγγίζει το άλλο δίδεται από την ακόλουθη εξίσωση (5):

$$v_{12} = \sqrt{8 k_b T / \pi m_{12}} \quad (5)$$

Όπου k_b η σταθερά Boltzmann και σχετικές μάζες $m_{12} = m_1 m_2 / (m_1 + m_2)$.

Υποθέτουμε ότι τα μόρια είναι ατάκτως διασκορπισμένα και ότι η κινητική ενέργεια τους είναι μεγαλύτερη του ενεργειακού χάσματος E_μ (E_μ η ενέργεια ένα ηλεκτρόνιο να εγκαταλείψει την στάθμη χαμηλότερης ενέργειας τους για να μεταβεί σε μια υψηλότερη όπου κυρίως μιλάμε για ηλεκτρόνια των τελευταίων στιβάδων των τροχιακών κατά τον σχηματισμό των μορίων). Θεωρούμε την πιθανότητα p_μ για κρούση των μορίων και συνεπώς και αντίδρασης θεωρώντας έναν κύλινδρο (ύψους $v_{12} * dt$, με εμβαδόν βάσης $A = \pi(r_1 + r_2)^2$) ως το χώρο που λαμβάνουν χώρα τα γεγονότα αυτά. Η σταθερά c_μ που περιγράφει την τάση των μορίων για αντίδραση δίδεται από τον ακόλουθη εξίσωση (6).

$$c_\mu = \Omega^{-1} \pi (r_1 + r_2)^2 + \left(8 k_b \frac{T}{\pi m_{12}} \right)^{-1/2} * p_\mu \quad (6)$$

Όπου η πιθανότητα είναι $p_\mu = e^{E_\mu / k_b T}$ σε έναν όγκο Ω .

Με γνώμονα την προηγούμενη ανάλυση, η πιθανότητα να πραγματοποιηθεί μια αντίδραση από το σύνολο των αντιδράσεων R_μ για την χρονική στιγμή $[t, t+dt]$ είναι $c_\mu * dt$ όπου η σταθερά c_μ σχετίζεται με τη γενική σταθερά που συναντούμαι στις χημικές εξισώσεις την k_i (βλέπε εξίσωση 7)



Πιο συγκεκριμένα η πιθανότητα που υπάρχει να αντιδράσουν 2 διαφορετικά στοιχεία X_i, X_{i+1} σε έναν όγκο Ω δίδεται από την ακόλουθη εξίσωση

$$a_\mu = X_i * X_{i+1} * k_i \quad (8)$$

Η όλη ανάλυση στηρίζεται σε Μαρκοβιανές μεθόδους όπου ο χρόνος της επόμενης αντίδρασης στηρίζεται στην πιθανότητα πραγματοποίησης της με βάση την εξίσωση 8 όπου η πιθανότητα αυτή είναι η $p(x, \tau | x, t)$. Η $1^{\text{η}}$ παράγωγος ως προς το χρόνο της πιθανότητας $p(x, \tau | x, t)$ περιγράφει την κατάσταση του συστήματος και αναφέρεται ως Γενική Χημική Εξίσωση (Chemical Master Equation) όπου περιγράφεται από τον ακόλουθο τύπο:

$$\frac{\partial P(x,t|x_0,t_0)}{\partial(t)} = \sum_{j=1}^n (a_j(x - v_j)P(x - v_j, t|x_0,t_0) - a_j(x)P(x,t|x_0,t_0)) \quad (9)$$

Η εξίσωση (9) δηλώνει με λεπτομέρεια την κατάσταση του βιοχημικού συστήματος την κάθε χρονική στιγμή δηλαδή περιγράφει την εξέλιξη ενός αντιδρώντος συστήματος στο χρόνο.

Η εξίσωση (8) περιγράφει την πιθανότητα που έχουν 2 διαφορετικά μεταξύ τους μόρια να αντιδράσουν. Όμως ανάλογα με το βαθμό της αντίδρασης δηλαδή με τον αριθμό των αντιδρώντων και το είδος των αντιδρώντων η εξίσωση (8) αναλύεται σε υπό περιπτώσεις όπως παρατηρείται και από τον πίνακα (1). Επίσης οι υπό-περιπτώσεις αυτές χαρακτηρίζουν και την σχέση του k_i με το c_μ . Δηλαδή στην περίπτωση που έχω 1^{ου} βαθμού αντίδραση το $k_\mu = c_\mu$ για 2^{ου} βαθμού ισχύει ότι $c_\mu = k_\mu / \Omega$ και για την περίπτωση 3^{ου} βαθμού του πίνακα 1 όπου η ίδια ουσία διασπάται θα έχω $c_\mu = 2 * k_\mu / \Omega$. Ακολούθως για τις υπόλοιπες περιπτώσεις θα έχω τους ανάλογους συσχετισμούς οι οποίοι και περιγράφονται στον πίνακα 1 με βάση την γενική εξίσωση (10).

$$a_j(x) = c_j * \frac{1}{2 * q} * [x_j * (x_j - 1) * (x_j - 2) * (x_j - (q - 1))] \quad (10)$$

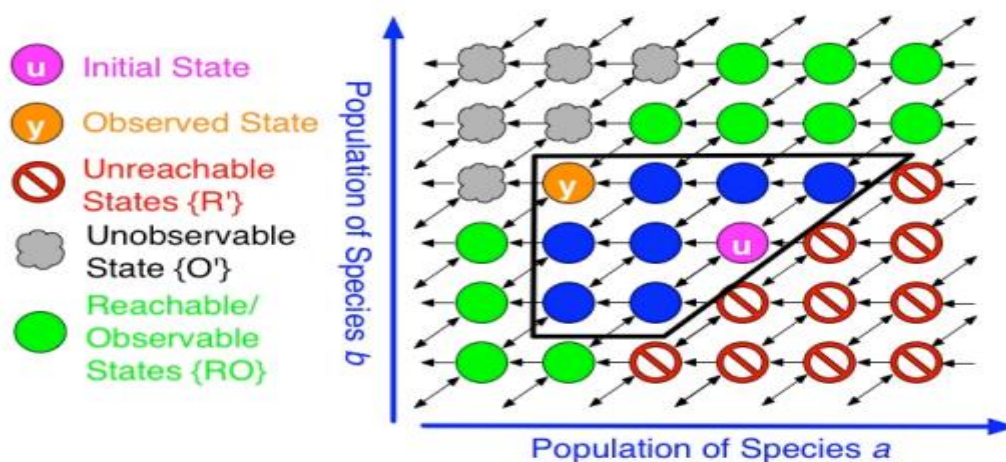
Πίνακας 1: Υπολογισμός όλων των περιπτώσεων της πιθανότητας αντίδρασης για αντιδράσεις μέχρι 3^{ου} βαθμού.

Reactions	Order	Propensity calculation
$X_1 \Rightarrow X_1 + X_2$	1 st	$k_i * [X_1]$
$X_1 + X_1 \Rightarrow X_2 + X_3$	2 nd	$(k_i * [X_1] * ([X_1] - 1) / 2)$
$X_1 + X_2 \Rightarrow X_3 + X_4$	2 nd	$(k_i * ([X_1] * [X_2]))$
$X_1 + X_2 + X_3 \Rightarrow X_1 + X_2 + X_5 + \dots X_n$	3 rd	$(k_i * ([X_1] * [X_2] * [X_3]))$
$X_1 + X_1 + X_2 \Rightarrow X_4 + X_6 + X_5 + \dots X_n$	3 rd	$(k_i * ([X_2] * ([X_1] - 1) * [X_1] * 0.5))$
$X_1 + X_1 + X_1 \Rightarrow X_4 + X_6 + X_5 + \dots X_n$	3 rd	$(k_i * ([X_1] * ([X_1] - 1) * ([X_1] - 2) * 0.16))$

2.2 Εισαγωγή στη στοχαστική προσομοίωση

Αποδεικνύεται ότι η Γενική Χημική εξίσωση (9) δεν μπορεί να λυθεί επακριβώς παρά μπορούμε να εξάγουμε με κάποιες υποθέσεις κάποιες συνιστώσες της κατάστασης του συστήματος. Μία προσεγγιστική μέθοδος για την επίλυση της γενικής εξίσωσης είναι οι διαφορικές εξισώσεις. Αυτό που γίνεται είναι ότι διαλέγουμε ένα μικρό χρονικό διάστημα Δt και διαφορίζουμε ως προς αυτό. Κατά συνέπεια η διαφορίση αυτή και εξαρτωμένου των συνθηκών και των συγκεντρώσεων των αντιδρώντων είτε θα ανανεώσει το σύστημα καθώς θα έχει πραγματοποιηθεί κάποια αντίδραση ή κατά τη χρονική διάρκεια αυτή δεν θα έχει ακόμα πραγματοποιηθεί κάποια μεταβολή-αντίδραση και απλά θα βρίσκεται το σύστημα αντιδράσεων σε μια φάση αναμονής. Παρατηρούμαι λοιπόν ότι όταν το Δt είναι αρκετά μικρό ναι μεν το σύστημα θα περιγραφεί λεπτομερώς αλλά θα υπάρχουν και

διαστήματα στα οποία δεν θα γίνονται καθόλου αντιδράσεις. Επίσης όλοι αυτοί οι μικροί βηματισμοί καθιστούν πολλές φορές απαγορευτική ως προς το χρόνο την προσομοίωση, γιατί η διαδικασία θα είναι εξαιρετικά αργή .



Σχήμα 13: Υπολογισμός εξέλιξης των στοιχείων της βιοχημικής αντίδρασης με βάση τη στοχαστική προσομοίωση.

Η στοχαστική προσομοίωση και πιο συγκεκριμένα οι αλγόριθμοι του Gillespie επιλύουν το πρόβλημα αυτό με μία μέθοδο η οποία καταργεί τις περιπτώσεις από τις οποίες στο σύστημα μου δεν γίνεται κάποια μορφής αντίδρασης και ουσιαστικά η όλη ανάλυση όπως θα δούμε υπολογίζει το Δt εκείνο για το οποίο θα έχω να γίνονται στο σύστημα μου διαρκώς αντιδράσεις. Με αυτόν τον τρόπο μειώνεται δραστικά ο χρόνος προσομοίωσης και διευκολύνεται ως ένα βαθμό η διαδικασία της ανάλυσης.

2.3 Περιγραφή Αλγορίθμων Στοχαστικών Προσομοιώσεων (Αλγόριθμοι Gillespie)

Πιο αναλυτικά η στοχαστική προσομοίωση με τους αλγόριθμους του Gillespie δεν επιλύει απευθείας την Γενική Χημική εξίσωση αλλά δημιουργεί μια νέα πιθανότητα με βάση την εξίσωση $P(\tau, \mu | x, t)$ για την πυροδότηση μιας αντίδρασης. Η ολοκλήρωση ως προς το χρόνο της εξίσωσης $P(\tau, \mu | x, t) d\tau$ εκφράζει ουσιαστικά την πιθανότητα πραγματοποίησης της επόμενης αντίδρασης από το σύνολο των αντιδράσεων R_μ σε χρόνο $[t + \tau, t + \tau + dt]$ με δεδομένο την κατάσταση του συστήματος $X(t) = x$.

Συμπεραίνουμε λοιπόν ότι ουσιαστικά μιλάμε για μια δεσμευμένη πιθανότητα δύο μεταβλητών, του χρόνου τ που είναι ο χρόνος που θα πραγματοποιηθεί η επόμενη αντίδραση και του δείκτη μ που δείχνει το ποια αντίδραση θα γίνει. Έτσι παραλείπονται τα σημεία εκείνα στα οποία δεν πραγματοποιείτε αντίδραση κάνοντας πιο γρήγορη την προσομοίωση. Η ανάλυση της *Direct method του Gillespie* που είναι πρόγονος της FRM δίδεται από τον ακόλουθο τύπο :

$$p(\tau, \mu|x, t) d\tau = Po(\tau|x, t) \times (\alpha_{\mu}(x)) d\tau \quad (11)$$

όπου $Po(\tau|x, t)$ εκφράζει την πιθανότητα να μην συμβαίνουν αντιδράσεις στο χρονικό διάστημα $[t, t + \tau)$ ενώ συμβαίνουν στο χρονικό διάστημα $[t + \tau, t + \tau + d\tau]$. Ισχύει ότι

$$Po(\tau + d\tau|x, t) = Po(\tau|x, t) \times [1 - \sum_{j=1}^m a_j(x)] \quad (12)$$

Παραγωγίζοντας ως προς το χρόνο έχουμε

$$\frac{d(Po(\tau + d\tau|x, t))}{d\tau} = -a_o(x) Po(\tau|x, t) \quad (13)$$

$$\text{Όπου } a_o = \sum_{j=1}^m a_j(x) \quad \text{και} \quad Po(\tau|x, t) = \exp(-a_o(x)\tau)$$

Συνεπώς η εξίσωση (13) ξαναγράφεται ως :

$$p(\tau, \mu|x, t) = \exp(-a_o(x)\tau) \times a_{\mu}(x) \quad (14)$$

ή

$$p(\tau, \mu|x, t) = a_o(x) \exp(-a_o(x)\tau) \times \frac{\alpha_{\mu}(x)}{\alpha_o(x)} \quad (15)$$

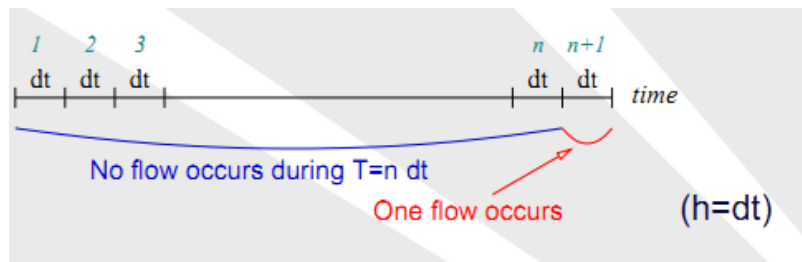
Άρα η εξίσωση (15) αποδεικνύει το γεγονός ότι η πιθανότητα $p(\tau, \mu|x, t)$ χωρίζεται στην πιθανότητα χρόνου τ και σε μία άλλη μ . Πιο συγκεκριμένα η μεταβλητή τ είναι μια τυχαία μεταβλητή (random number) με μέσο όρο και τυπική απόκλιση $1/\alpha_o(x)$ όπως φέρεται από το 1^ο κομμάτι της εξίσωσης (15). Ο αριθμός μ είναι μια τυχαία μεταβλητή (random number) με πιθανότητα $\frac{\alpha_{\mu}(x)}{\alpha_o(x)}$.

2.4 Περιγραφή του Αλγορίθμου του Gillespie Direct Method

Με γνώμονα αυτές τις παρατηρήσεις ο Gillespie δημιούργησε αρχικά τους θεμελιώδους αλγορίθμους τον *Direct SSA* και τον *First Reaction Method*. Για το Direct Method αρχικά ο αλγόριθμος ξεκινά με δεδομένες κάποιες αρχικές συνθήκες όπως ο χρόνος $t(0)$ και αρχικές

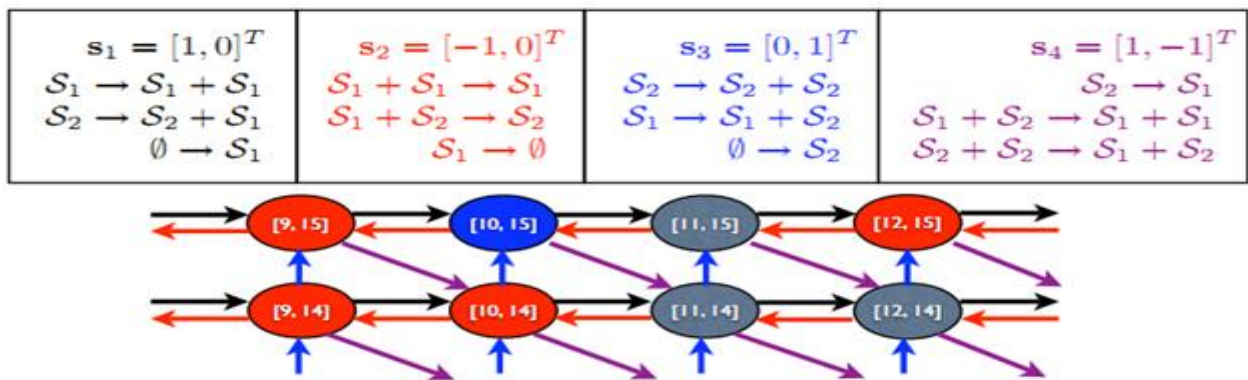
συγκεντρώσεις $X_j(0)$. Αρχικά υπολογίζεται για όλες τις αντιδράσεις η πιθανότητα πραγματοποίησης τους που εξαρτάται όπως δείξαμε από τις συγκεντρώσεις και το k ρυθμός μεταβολής (reaction rate) δηλαδή υπολογίζουμε το $a_j(x) = X_j * X_{j+1} * k_j$ και το άθροισμα όλων των πιθανοτήτων $a_j(x)$ που είναι το a_0 όπως δείξαμε στις εξισώσεις (11-15), βέβαια αυτό αποτελεί και το μειονέκτημα της Direct μεθόδου του Gillespie γιατί υπάρχει το ενδεχόμενο να μην αλλάζουν όλες οι πιθανότητες που σημαίνει ότι θα καταναλωθεί υπολογιστική ισχύς χωρίς ουσία. Έπειτα θα πρέπει να επιλεγούν 2 τυχαίοι αριθμοί r_1 και r_2 στο διάστημα $[0,1]$. Αυτοί οι 2 αριθμοί όπως αναλύθηκε και προηγουμένως χρησιμοποιούνται για τον υπολογισμό των ελάχιστων χρονικών στιγμών t_{min} και των μ πραγματοποιήσιμων αντιδράσεων. Έτσι γίνεται ο υπολογισμός του χρόνου τ δηλαδή της αντίδρασης που θα γίνει με βάση τον ακόλουθο γενικό τύπο (16):

$$\int_0^{\tau} a_0(x) \exp(-a_0(x) \tau') d\tau' = 1 - r_1 \quad (16)$$



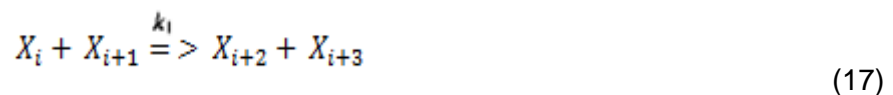
Σχήμα 14: Στιγμιότυπα χρόνου αντίδρασης.

Η μεταβλητή μ υπολογίζεται αθροίζοντας τα a_j και έχοντας υπολογίσει το άθροισμα όλων των $a_j(x)$ μέχρι το συγκεκριμένο j ώστε $\sum_{j=1}^{\mu} a_j(x) > r_2 * a_0$, οπότε μόλις εκπληρωθεί η προηγούμενη συνθήκη επιλέγεται η παράμετρος μ που δείχνει το ποια αντίδραση θα πραγματοποιηθεί. Το επόμενο βήμα είναι να αυξήσουμε το βήμα χρόνου προσθέτοντας στο παλιό χρόνο την ευρισκόμενη ελάχιστη τιμή $t_{new} = t_{old} + \tau$ και ανάλογα με την παράμετρο μ του προηγούμενου βήματος που αντιστοιχεί στη αντίδραση που θα γίνει θα βρεθούν οι νέες συγκεντρώσεις με βάση της στοιχειομετρία της επιλεχθείσας αυτής αντίδρασης που πραγματοποιήθηκε .

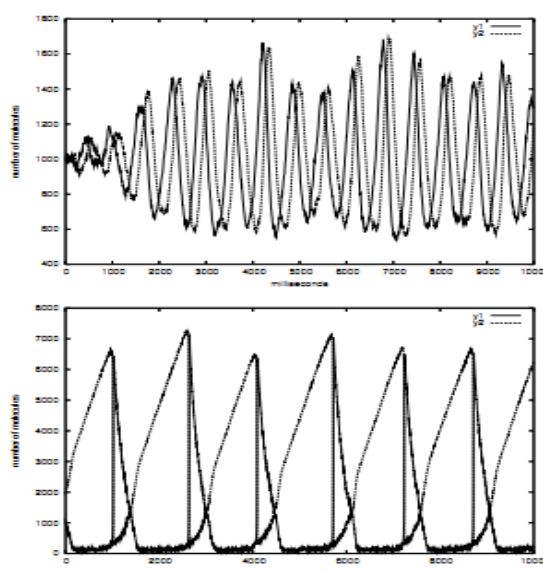


Σχήμα 15: Περιγραφή της εξέλιξης μιας βιοχημικής αντίδρασης με βάση τη στοιχειομετρία.

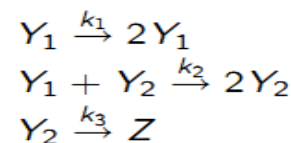
Με αυτόν τον τρόπο και με γνώμονα τις αρχικές συνθήκες (αρχικές συγκεντρώσεις και χρόνος έναυσης των αντιδράσεων) καθώς επίσης και με δεδομένες τις αντιδράσεις που είναι να πραγματοποιηθούν καθώς και τις στοιχειομετρίες τους μπορούμε να υπολογίζουμε την εξέλιξη συστημάτων $1^{ou}, 2^{ou}$, και ανωτέρου βαθμού (συνήθως ο βαθμός αντίδρασης σε βιοχημικές αντιδράσεις είναι μέχρι 3^{ou}) της μορφής που φαίνεται στην εξίσωση (17)



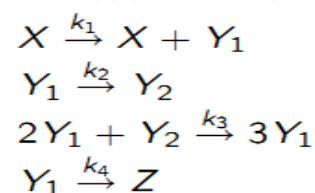
Η εξίσωση (17) δείχνει μια Βιοχημική αντίδραση 2^{ou} βαθμού λόγο των 2 αντιδρώντων X_j, X_{j+1} . Στο σχήμα 16 μπορούμε να δούμε παραδείγματα στοχαστικών προσομοιώσεων από τυπικά πραγματικά μοντέλα απλών συστημάτων που πραγματοποιούν ταλαντώσεις λόγο ανάδρασης των συγκεντρώσεων των στοιχείων τους από τις βιοχημικές αντιδράσεις του δικτυώματος.



Lotka reactions



Brusselator reactions



Σχήμα 16: Παραδείγματα Στοχαστικών προσομοιώσεων.

Θα αρχίσουμε την περιγραφή όλων των αλγορίθμων στοχαστικής προσομοίωσης από την

Direct Μέθοδο του Gillespie. Ο αλγόριθμος 1 περιγράφει αναλυτικά όλα τα βήματα της *Direct Method* του Gillespie που είναι ο προκάτοχος της FRM μεθόδου. Αρχικά υπολογίζονται οι πιθανότητες αντίδρασης $a_j(x)$ και αθροίζονται. Μετά επιλέγονται 2 τυχαίοι αριθμοί r_1 και r_2 . Κατόπιν υπολογίζεται ο χρόνος με βάση την εξίσωση (19). Έπειτα υπολογίζεται η αντίδραση που θα πυροδοτηθεί την επόμενη χρονική στιγμή με βάση την συνθήκη της εξίσωση 20. Με γνώμονα την αντίδραση που πραγματοποιήθηκε και με βάση τη στοιχειομετρία της αντίδρασης αυτής βρίσκονται οι καινούργιες συγκεντρώσεις.

Περιγραφή DM αλγορίθμου (Gillespie's Direct Method)

Algorithm 1. First Reaction method

Initialize : $t = t_0$ and $x = x_0$.

Loop:

{ Evaluate propensity functions $a_j(x)$ at state x , and also their sum

$$a_0 = \sum_{j=1}^m a_j(x) \quad (18)$$

Draw two unit uniform random numbers, r_1 and r_2

Determine the time, τ , until the next reaction:

$$\tau = \frac{1}{a_0(x)} * \ln \left(\frac{1}{r_1} \right) \quad (19)$$

Determine the next reaction, R_μ where μ = the smallest integer satisfying

$$\sum_{j=1}^{\mu} (a_j(x)) > r_2 * a_0(x) \quad (20)$$

Determine the new state after reaction μ : $t = t + \tau$ and $x = x + \nu_\mu$.

if t is greater than the desired simulation time then halt.

else

Go to Loop };

2.5 Περιγραφή FRM Αλγορίθμου, υπολογιστική πολυπλοκότητα

Η DM μέθοδος του Gillespie λειτουργεί πολύ καλά για μικρά συστήματα βιοχημικών αντιδράσεων. Το πρόβλημα έγκειται όταν το βιοχημικό σύστημα είναι μεγάλο, λόγω του μεγάλου υπολογιστικού χρόνου της μεθόδου αυτής. Για την επίλυση αυτού του προβλήματος ο Gillespie εφάρμοσε μια νέα μέθοδο στηριζόμενη στην DM. Η μέθοδος αυτή υπολογίζει για την κάθε R_j αντίδραση το χρόνο τ_j με βάση την εξίσωση (21) που φαίνεται στην περιγραφή του Αλγορίθμου 2. Ο μικρότερος τ_j χρόνος κερδίζει και η αντίδραση που αντιστοιχεί στο χρόνο αυτό θα εκτελεστεί πρώτη. Ουσιαστικά βλέπουμε ότι καταργούνται οι εξαρτήσεις κάποιων βημάτων (εξισώσεις 18-20) το οποίο επιταχύνει τους χρόνους υπολογισμού και επιτρέπει την παραλληλοποίηση του αλγορίθμου.

Περιγραφή FRM αλγορίθμου

Algorithm 2. First Reaction method

Initialize $X(t_0) = x_0$.

While {True} {

For every j do % start a new reactions cycle

{ Evaluate propensity function ($a_j(x)$)

Determine the time τ_j until the next R_j reaction:

$$\tau_j = \frac{1}{a_j} * \ln\left(\frac{1}{r_j}\right) \quad (21)$$

where r_j is a unit uniform random number

End for (j) }

Let R_μ be the “winning” reaction with the smallest τ_j

Determine the new state after reaction R_μ occurs

$t = t + \tau_\mu$ and $X(t + \tau_\mu) = x + \nu_\mu$.

If t exceeds the desired simulation time, halt

End While{True} }

2.6 Ανάλυση NRM Αλγορίθμου, κίνητρα για την δημιουργία του αλγορίθμου και υπολογιστική πολυπλοκότητα

Αναλύοντας περαιτέρω τον FRM αλγόριθμο διαπιστώθηκε από τους Gibson και Brucks ότι μπορεί να επιταχυνθεί ακόμα περισσότερο η διαδικασία του FRM. Έτσι δημιουργήθηκε ο αλγόριθμος *Next Reaction Method* ο οποίος σε σχέση με τον FRM θεωρώντας ότι έχει πολυπλοκότητα m επιταχύνεται με πολυπλοκότητα $\log(m)$. Η επιτάχυνση αυτή οφείλεται στον γράφο εξαρτήσεων της NRM μεθόδου. Ο γράφος αυτός εξαρτήσεων προκύπτει από το γεγονός ότι δεν επηρεάζονται όλες οι αντιδράσεις από την πραγματοποίηση της αντίδρασης που έδωσε τον ελάχιστο χρόνο. Δηλαδή όπως είναι φυσικό κάποιες αντιδράσεις επηρεάζουν κάποιες άλλες (έχουν κοινά στοιχεία συγκεντρώσεων στα οποία συμμετέχουν) και κάποιες όχι. Για παράδειγμα στις αντιδράσεις της μορφής R_1, R_2, R_3 που βλέπουμε παρακάτω, εάν η R_1 δημιουργεί με βάση την εξίσωση (21) τον μικρότερο χρόνο εκτέλεσης άρα και θα εκτελεστεί πρώτη δεν θα επηρεάσει την R_2 αλλά την R_3 αφού τα στοιχεία της έχουν σχέση με τα αντιδρώντα της R_3 και άρα επηρεάζουν το παράγωγο ($a_j(x)$) για $j = 3$ αυτής, όπου με βάση την εξίσωση 21 θα επηρεάσουν και το χρόνο εκτέλεσης.

- R1: $X_1 + X_2 = k_1 > X_3 + X_4$
- R2: $X_5 + X_6 = k_2 > X_7 + X_8$
- R3: $X_3 + X_4 = k_3 > X_9 + X_5$

Με αυτόν τον τρόπο και φτιάχνοντας ένα γράφο εξαρτήσεων μπορούμε να αποφύγουμε το γεγονός ότι σε κάθε βήμα πρέπει να υπολογίζονται κάθε φορά όλα τα $a_j(x)$ και $\tau_j(x)$ αλλά υπολογίζονται μόνο οι πιθανότητες των αντιδράσεων που αλλάζουν $a_\mu(x)$ καθώς και οι χρόνοι $\tau_\mu(x)$ με αποτέλεσμα την επιτάχυνση της διαδικασίας. Πιο αναλυτικά ο Αλγόριθμος 3 περιγράφει τα βήματα του NRM .

Περιγραφή NRM αλγορίθμου

Algorithm 3. Next Reaction method

Initialize : $t = t_0$ and $x = x_0$.

Generate a dependency graph, G .

Evaluate propensity functions $a_j(x)$ at state x .

For each j , determine the time, τ_j , until the next R_j reaction:

$$\tau_j = \frac{1}{a_j} * \ln \left(\frac{1}{r_j} \right) \quad (22)$$

where each r_j is a unit uniform random number.

Loop: {

Store the τ_j values in an indexed priority queue Q .

Let R_μ be the reaction whose τ_μ is the smallest.

Let τ equal τ_μ

Determine the new state after reaction R_μ : $t = t + \tau$ and $x = x + \nu_\mu$.

For each edge (μ, α) in the dependency graph G ,

Set $a_{old} = a_\alpha$ and update a_α .

If $\alpha \neq \mu$, set $\tau_\alpha = (a_{old} / a_\alpha) (\tau_\alpha - t) + t$. (23)

If $\alpha = \mu$, generate a random number, r_μ , and

$$\tau_\mu = t + \frac{1}{a_\mu} * \ln \left(\frac{1}{r_\mu} \right) \quad (24)$$

If t is greater than the desired simulation time then halt.

Record (x, t) go to Loop}.

Αρχικά παρατηρούμαι ότι η διαδικασία είναι ίδια με τον FRM μέχρι και την 1^η καταγραφή όλων των $a_j(x)$ και $\tau_j(x)$ μετά ο αλγόριθμος εξελίσσεται με βάση τον γράφο εξαρτήσεων.

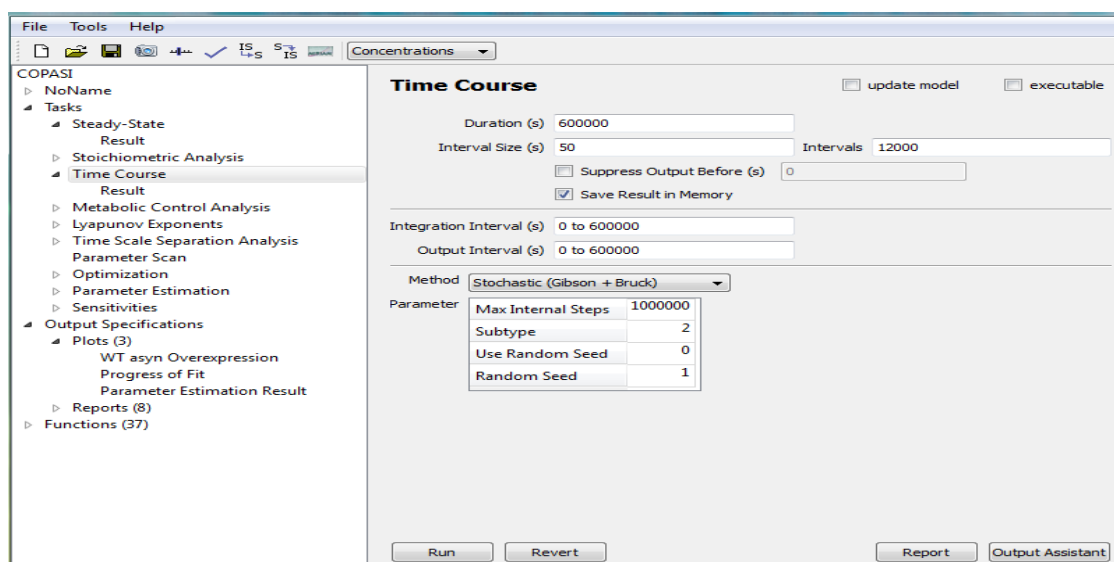
Ο NRM με αυτόν τον τρόπο αποδεικνύεται γρηγορότερος αλγόριθμος από τους υπόλοιπους όμως υπάρχει μεγάλη δυσκολία στην παραλληλοποίηση του λόγω του γράφου εξαρτήσεων έτσι ώστε για πολύ μεγάλα μοντέλα να γίνεται συγκρίσιμος και να αποδεικνύεται ποιο εύχρηστη η χρήση του FRM.

Στην σχεδίαση μας με το κύκλωμα FRM4X_NRM4X+Leon κατορθώσαμε να παραλληλοποιήσουμε το γράφο εξαρτήσεων άρα και την διαδικασία με αποτέλεσμα την βελτιστοποίηση του αλγορίθμου για μεγάλα μοντέλα βιοχημικών δικτύων. Επίσης κατορθώσαμε με παραλληλοποίηση σε μεγάλη κλίμακα να βελτιστοποιήσουμε και να επιταχύνουμε την εύρεση του μικρότερου χρόνου σε δίκτυα μεγάλων αντιδράσεων το

οποίο επιταχύνει περεταίρω τον NRM αλγόριθμο καθώς το κρίσιμο μονοπάτι της διαδικασίας μεταφέρετε εκεί. Όμως ένα βασικό μειονέκτημα είναι οι απαιτήσεις σε υλικό (hardware) στο FPGA σε σύγκριση με τον FRM. Επίσης παρατηρήθηκε και μείωση της συχνότητας λειτουργίας λόγω των πολλαπλών ανταλλαγών δεδομένων με τις εσωτερικές μνήμες Block Rams το οποίο όμως μειονέκτημα υπερκαλύπτεται από το γεγονός ότι ελαχιστοποιείτε ο χρόνος προσομοίωσης λόγω των εξαρτήσεων. Βέβαια η αρχιτεκτονική αυτή είναι καινούργια και πρέπει να υπάρξει περεταίρω έλεγχος (testing) της μεθόδου και (validation) ορθότητας των αποτελεσμάτων αυτής σε σύγκριση πάντα με την υλοποίηση μας με τις FRM σχεδιάσεις.

2.7 Γενική περιγραφή των μεθόδων και των εργαλείων προσομοίωσης Simulation tools (Copasi, iBioSim).

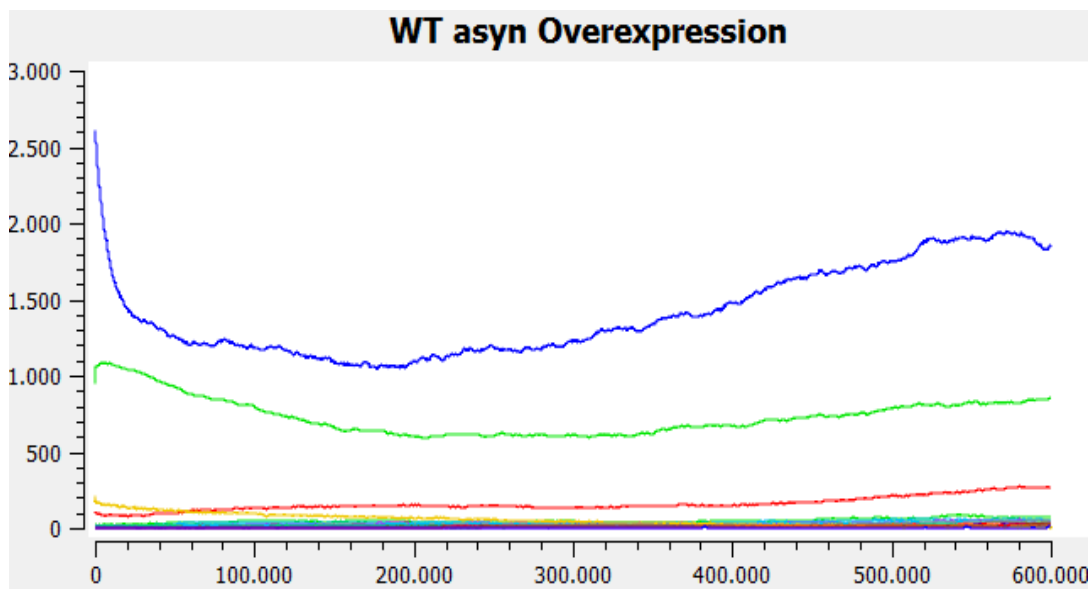
Με βάση όλους αυτούς τους στοχαστικούς αλγόριθμους που αναλύσαμε, διάφοροι προσομοιωτές σε Software έχουν υλοποιηθεί. Δύο από τους πιο γνωστούς και ευρέως διαδεδομένους είναι το Copasi[11] και το Ibiosim tool[12]. Οι προσομοιωτές αυτοί χρησιμοποιούν μοντέλα από την SBML βιβλιοθήκη [13] και χρησιμοποιούν αλγορίθμους γραμμένους σε C++ για το Copasi ή Java για το Ibiosim tool για την πραγματοποίηση των στοχαστικών προσομοιώσεων. Έτσι ο χρήστης θέτει το χρόνο προσομοίωσης και την μέθοδο που θέλει και αρχίζει η διαδικασία προσομοίωσης. Με βάση αυτά τα εργαλεία επικυρώσαμε την ορθότητα των κυκλωμάτων μας συγκρίνοντας τις διάφορες κυματομορφές .



Σχήμα 17: Προσομοιωτής Copasi.

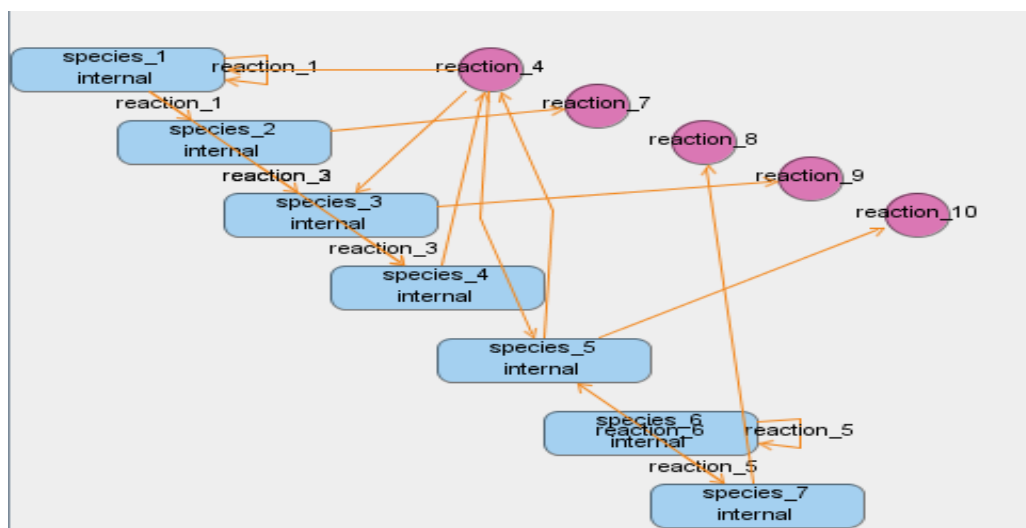
Το σχήμα 18 δείχνει ένα παράδειγμα προσομοίωσης με το εργαλείο Copasi για μια προσομοίωση με την NRM μέθοδο του Gibson and Brucks. Επίσης το εργαλείο αυτό δίνει

και την ανάλυση της στοιχειομετρίας από ένα SBML μοντέλο καθώς επίσης ο χρήστης έχει και την δυνατότητα δημιουργίας ενός SBML μοντέλου και την εξαγωγή απευθείας του xml αρχείου. Παρακάτω φαίνεται ένα παράδειγμα μιας κυματομορφής από προσομοίωση ενός πειραματικού μοντέλου για την ασθένεια του Parkinson όπως προσομοιώθηκε από το συγκεκριμένο προσομοιωτή.

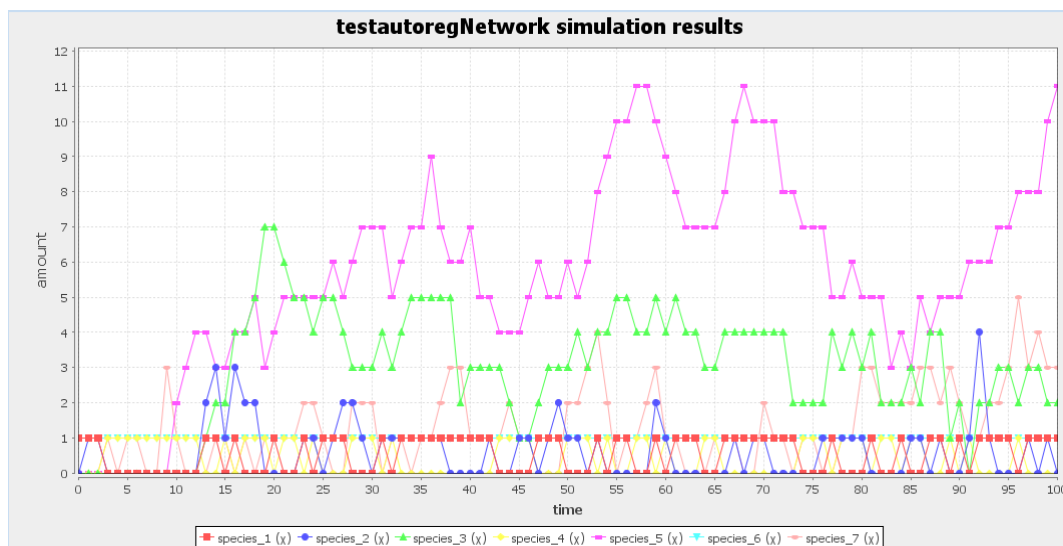


Σχήμα 18: Παράδειγμα Στοχαστικής προσομοίωσης με το Copasi.

Παρόμοιο εργαλείο είναι και το Ibiosim tool το οποίο επιτρέπει σε σχέση με το Copasi την διεξαγωγή πολλαπλών προσομοιώσεων και την εύρεση της μέσης τιμής από όλες τις προσομοιώσεις του συστήματος. Επίσης επιτρέπει και την γραφική απεικόνιση καθώς εμφανίζει και διάγραμμα εξαρτήσεων των αντιδράσεων το οποίο φαίνεται παρακάτω.



Σχήμα 19: Απεικόνιση δικτύματος αντιδράσεων και αλληλεξαρτήσεων από τον προσομοιωτή Ibiosim tool.



Σχήμα 20: Παράδειγμα προσομοίωσης με το εργαλείο Ibiosim.

Αποδείχθηκε ότι οι προσομοιώσεις αυτές από τα συγκεκριμένα εργαλεία όταν απαιτείτε μεγάλος όγκος δεδομένων κρίνονται τρομερά αργές παρόλο τις απλοποιήσεις τις οποίες επιδέχονται καθιστώντας έως και αδύνατη πολλές φορές την προσομοίωση. Επίσης φαίνεται και τρομερά δύσκολη η καταγραφή σε κάποιο αρχείο της μεταβολής των στοιχείων σε σύγκριση πάντα με το δικό μας σύστημα και ειδικότερα όταν ο όγκος των δεδομένων αυξάνει. Επιπλέον σε όλα τα εργαλεία φαίνεται ότι ο χρήστης ειδικότερα για μεγάλο όγκο πληροφορίας δεν μπορεί να δει το πως κυμαίνονται οι συγκεντρώσεις των στοιχείων την κάθε χρονική στιγμή το οποίο κρίνεται ιδιαίτερα αναγκαίο στην ανάλυση μεγάλων μοντέλων βιοχημικών αντιδράσεων. Ένα άλλο μειονέκτημα το οποίο φάνηκε στην ανάλυση πολύπλοκων βιολογικών μοντέλων και για μεγάλο χρόνο προσομοίωσης καθώς και στις πολλαπλές προσομοιώσεις είναι το γεγονός ότι το σύστημα υπολογιστή-προγράμματος προσομοίωσης δεν μπορεί να ανταπεξέλθει στον τόσο μεγάλο όγκο δεδομένων με αποτέλεσμα να βγαίνει η διαδικασία όλη εκτός μνήμης (out of memory). Φαίνεται λοιπόν η ανάγκη για ύπαρξη μιας hardware υλοποίησης η οποία θα επιτρέπει την ανάλυση μεγάλου όγκου δεδομένων χωρίς προβλήματα.

ΚΕΦΑΛΑΙΟ 3

ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΤΩΝ FRM SoCs

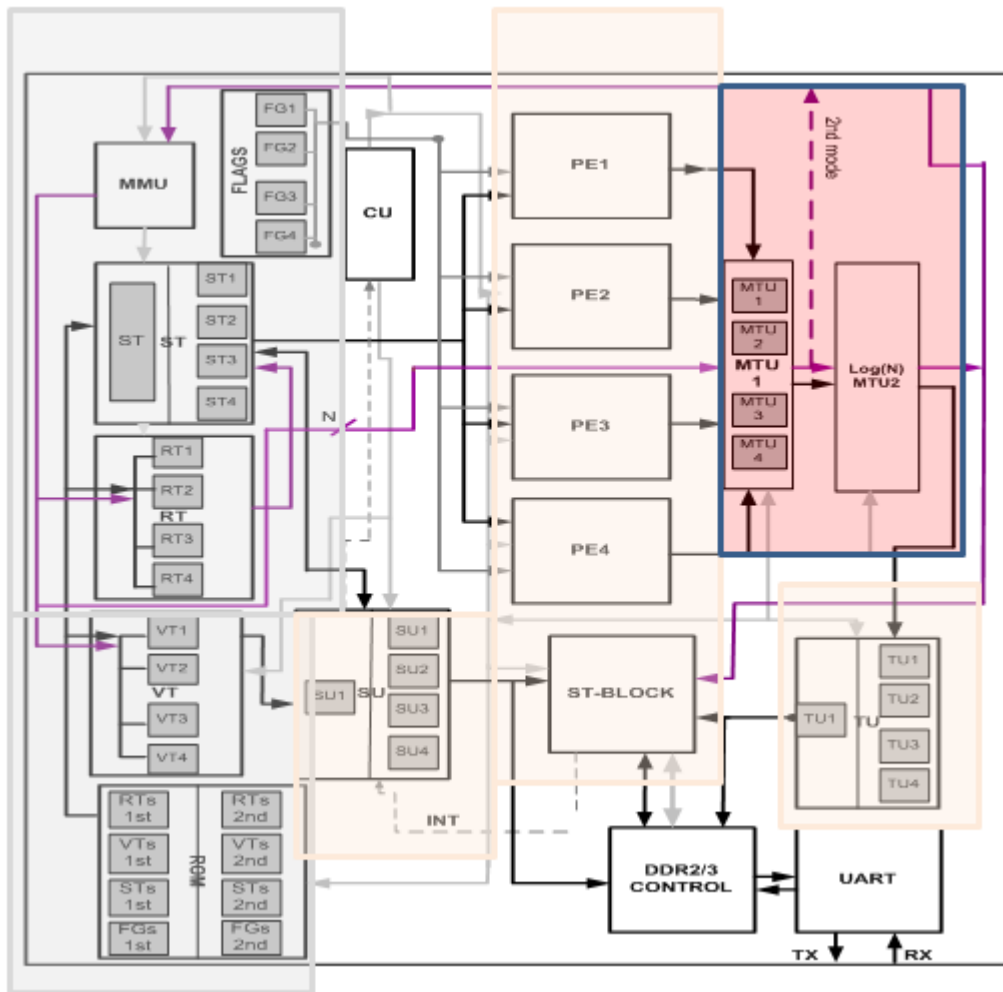
3.1 Εισαγωγή στην αρχιτεκτονική των FRM SoC

Το κεφάλαιο αυτό περιγράφει και αναλύει την αρχιτεκτονική των συστημάτων που υλοποιούν τον αλγόριθμο FRM. Πιο συγκεκριμένα θα γίνει ανάλυση όλων των υπό μονάδων και της λειτουργίας τους, της διαδρομής των δεδομένων καθώς και του τρόπου προγραμματισμού της μονάδας. Επίσης θα γίνει αναφορά στον τρόπο παραλληλοποίησης και παραμετροποίησης της μονάδας, ανάλογα με τις απαιτήσεις του εκάστωτε biomodel προς ανάλυση. Πιο αναλυτικά όλες οι αρχιτεκτονικές που θα περιγραφούν είναι αναδομήσιμες (reconfigurable) και παραμετροποιημένες (parametric). Η παραμετροποίηση έγκειται ως προς τον αριθμό των αντιδράσεων και των στοιχείων που υπάρχουν σε μια αντίδραση καθώς και του βαθμού της αντίδρασης (reaction order) 1^{ος}, 2^{ος}, 3^{ος} βαθμός κλπ. Όλες οι σχεδιάσεις όπως θα δούμε είναι αναδομήσιμες (reconfigurable) μέσω του συστήματος SysPy. Όπως θα αναλύσουμε το SysPy είναι ένα ερευνητικό πρόγραμμα που επιτρέπει τη δημιουργία VHDL κώδικα για την δημιουργία SSA-SoC ειδικού σκοπού εξαρτώμενου από το εκάστωτε biomodel προς προσομοίωση και τις ανάγκες του χρήστη.

Το παρακάτω σχήμα δείχνει την αρχιτεκτονική του κυκλώματος FRM4X δηλαδή με 4 επεξεργαστές PE. Χαρακτηριστικό στοιχείο της σχεδίασης είναι ότι η συγκεκριμένη αρχιτεκτονική υποστηρίζει παράλληλες προσομοιώσεις αλλά και ανεξάρτητες παράλληλες προσομοιώσεις χρησιμοποιώντας όλο το εύρος των επεξεργαστών PE (Processing Elements).

Παρατηρούμαι από την σχήμα 21 ότι η αρχιτεκτονική είναι χωρισμένη σε διάφορα μέρη όπως παρατηρείται από τα έγχρωμα τετράγωνα. Αυτό έγινε για την προοπτική υλοποίησης στην μονάδα παραπάνω από έναν αλγόριθμο στοχαστικής προσομοίωσης και έτσι να μοιράζονται τις βασικές υπό μονάδες το οποίο εξοικονομεί χώρο στο FPGA.

3.2 Γενική περιγραφή του FRM4X και όλων των υπό-μονάδων



Σχήμα 21: Ανάλυση διαγράμματος αρχιτεκτονικής FRM διαχωρισμένο σε τομείς με N=4 επεξεργαστές.

Με βάση το σχήμα 21 παρατηρούμαι με γκρι πλαίσιο την μονάδα οργάνωσης των μνημών (memory subsystem) η οποία περιλαμβάνει όλες τις μνήμες και την μονάδα διαχείρισης της διευθυνσιοδότησης (MMU). Επίσης με πορτοκαλί πλαίσιο φαίνεται η μονάδα αριθμητικής ανάλυσης (DSP unit) που έχει αριθμητικές μονάδες στηριζόμενες στην αριθμητική κινητής υποδιαστολής με βάση το πρωτόκολλο IEEE 754 (floating point single precision 32 bit). Η μονάδα αριθμητικής ανάλυσης (DSP unit) δέχεται το κυρίως επεξεργαστικό φορτίο και αποτελεί το κρίσιμο μονοπάτι της σχεδίασης μας, καθώς καθορίζει τη συχνότητα λειτουργίας του συστήματος. Με κόκκινο πλαίσιο φαίνεται η μονάδα εύρεσης του ελαχίστου χρόνου (Minimum Time Unit) όπου πρόκειται για μια δένδροειδής δομή πολλαπλών συγκριτών οι οποίοι αναλαμβάνουν την εύρεση του ελαχίστου χρόνου από όλους τους επεξεργαστές. Η εύρεση του ελαχίστου χρόνου όπως αναλύσαμε και στα προηγούμενα κεφάλαια εκφράζει το ποιά αντίδραση θα γίνει στο συγκεκριμένο χρονικό διάστημα. Η συγκεκριμένη μέθοδος υπολογισμού ελαχίστου χρόνου εμφανίζεται σχεδόν σε όλες τις παραλλαγές των στοχαστικών αλγορίθμων.

Σε όλα τα συστήματα υπάρχει και η μονάδα ελέγχου η οποία ελέγχει και συντονίζει όλες τις υπό-μονάδες. Επίσης υπάρχει και η δυνατότητα προγραμματισμού της μονάδας ελέγχου και συνεπώς όλου του συστήματος από τον εσωτερικό καταχωρητή ελέγχου (control register) ο οποίος προγραμματίζεται ανάλογα με τις απαιτήσεις του χρήστη και συνεπώς προγραμματίζει και την μονάδα ελέγχου η οποία με τη σειρά της ενεργοποιεί τις κατάλληλες υπό μονάδες για να γίνει η απαιτούμενη εργασία.

Πιο αναλυτικά την μονάδα οργάνωσης μνημών (memory subsystem) την αποτελούν η ST, (Species Table) μνήμη κράτησης συγκεντρώσεων στοιχείων αντίδρασης, VT (μνήμη με την στοιχειομετρία για την κάθε αντίδραση), RT (Reaction Table) μνήμη κράτησης δεικτών (pointers στο Species Table) που διαμορφώνουν τις αντιδράσεις. Επίσης αυτή η μονάδα κρατάει και τον παράγοντα k (reaction rate) της αντίδρασης. Η μνήμη Flag Table είναι η μονάδα προγραμματισμού για τον υπολογισμό της πιθανότητας αντίδρασης και η MMU (Memory Management Unit) είναι η μονάδα διευθυνσιοδότησης υπεύθυνη για την διευθυνσιοδότηση όλων των προαναφερθέντων μνημών).

Την μονάδα αριθμητικής ανάλυσης (DSP unit) την αποτελούν οι κάτωθι υπό μονάδες: Οι επεξεργαστές PE (processing elements) οι οποίοι υπολογίζουν το χρόνο $\tau_j(x)$ με βάση την εξίσωση 21 κεφάλαιο 2 καθώς και την πιθανότητα αντίδρασης $\alpha_j(x)$ με βάση την εξίσωση 8 του κεφαλαίου 2.

Επίσης όπως παρατηρούμαι υπάρχει η μονάδα SU (species update) καθώς και TU (time update). Η μονάδα SU ουσιαστικά διαμορφώνει τις συγκεντρώσεις των στοιχείων που συμμετέχουν σε μια βιοχημική αντίδραση ανάλογα με την στοιχειομετρία της εκάστως αντίδρασης που πραγματοποιείτε. Η μονάδα TU διαμορφώνει την εξέλιξη του χρόνου αντίδρασης με βάση το μικρότερο χρόνο που υποδεικνύει το ποιά αντίδραση θα γίνει δηλαδή $t = t_{old} + \tau_{min}$. Επίσης υπάρχει πρόβλεψη και για την διεξαγωγή στατιστικών στοιχείων απευθείας από τη μονάδα (St-block) η οποία υπολογίζει το μέσο όρο και τις τυπικές αποκλίσεις για τις συγκεντρώσεις των στοιχείων (species) την κάθε χρονική στιγμή για το εκάστως βιοχημικό μοντέλο.

Η μονάδα εύρεσης του ελαχίστου χρόνου (Minimum Time Unit) περιλαμβάνει τις υπό-μονάδες MTU1, και $\log(N)$ MTU2 όπου η MTU1 βρίσκει σειριακά το μικρότερο χρόνο από τα δεδομένα που έρχονται από τον κάθε επεξεργαστή και η μονάδα MTU2 που αποτελείτε από μια δενδροειδής δομή συγκριτών υπολογίζει την ελάχιστη τιμή χρόνου αντίδρασης από όλους τους επεξεργαστές δηλαδή το συνολικό ελάχιστο χρόνο.

Επίσης μπορούμε να διακρίνουμε και κάποια συστήματα επικοινωνίας που

χρησιμοποιούνται όταν η μονάδα λειτουργεί αυτόνομα (χωρίς το Interface με τον χρήστη μέσω του επεξεργαστή Leon) και αυτά είναι ο εσωτερικός DDR2 controller που αποθηκεύει τα δεδομένα στην εσωτερική DDR2 μνήμη του FPGA καθώς και μονάδα σειριακής επικοινωνίας UART, για την ανταλλαγή δεδομένων με το χρήστη αλλά και για τον προγραμματισμό της μονάδος από εντολές που μπορεί να δώσει σειριακά ο χρήστης. Τα δεδομένα μέσω σειριακής θα μπορούν να εξαχθούν μέσω του Matlab για να γίνει από εκεί η περαιτέρω επεξεργασία.

3.3 Ανάλυση τρόπων λειτουργίας για N ξεχωριστές προσομοιώσεις και μία N παράλληλη προσομοίωση

Όπως αναφέραμε και προηγουμένως η αρχιτεκτονική αυτή επιτρέπει την διεξαγωγή για μεγάλα δίκτυα αντιδράσεων μίας N παράλληλης προσομοίωσης όπου διαμοιράζονται οι αντιδράσεις δια του αριθμού N των επεξεργαστών (PEs) καθώς επίσης μπορούν να γίνουν N παράλληλες προσομοιώσεις για την διεξαγωγή στατιστικών συμπερασμάτων για το ίδιο το βιοχημικό μοντέλο. Αναλυτικότερα μπορεί να υπάρξει η ανάγκη μελέτης του ίδιου βιοχημικού μοντέλου με παραλλαγές είτε των αρχικών συγκεντρώσεων είτε του αριθμού k (reaction rate). Με αυτού του είδους αρχιτεκτονικής επιταχύνεται η παραπάνω διαδικασία πολύ γρήγορα και εύκολα.

Στον διαμοιρασμό των αντιδράσεων σε περίπτωση μίας N παράλληλης προσομοίωσης όπως θα δούμε και παρακάτω αυξάνεται η απόδοση του συστήματος μας (throughput) και μειώνεται ο χρόνος προσομοίωσης πράγμα το οποίο κρίνεται αναγκαίο σε πολύ μεγάλα δίκτυα βιοχημικών αντιδράσεων.

Θα αρχίσουμε την ανάλυση μας αρχικά για την διεξαγωγή μίας παράλληλης προσομοίωσης και κατόπιν θα συνεχίσουμε με τον 2^ο τρόπο λειτουργίας δηλαδή τις N πολλαπλές ανεξάρτητες προσομοιώσεις.

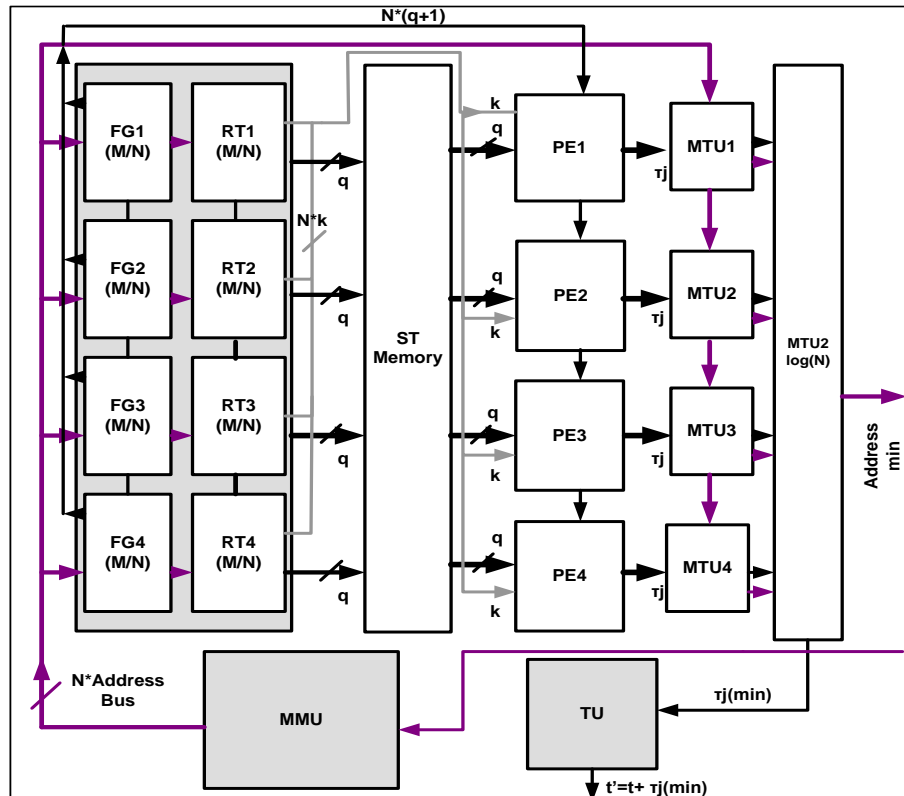
Αρχικά αρχίζει η διαμόρφωση των μνημών με βάση το εκάστωτε SBML μοντέλο, δηλαδή ανάλογα το μοντέλο και μέσω διαφόρων προγραμμάτων (scripts) σε Python δημιουργείτε η αρχικοποίηση των μνημών, όπου δημιουργείται ένας VHDL κώδικας ο οποίος έχει αρχικοποιημένες τις μνήμες με πληροφορίες που συλλέγονται από το SBML αρχείο που περιγράφει το βιοχημικό μοντέλο. Σε αυτό το σημείο πρέπει να αναφερθεί ότι η SBML βιβλιοθήκη διαθέτει ειδικούς μετατροπείς (parsers) που μπορούν μέσω κώδικα σε C να εξαγάουν απευθείας από το xml αρχείο τις απαραίτητες πληροφορίες, όπως είναι οι συγκεντρώσεις στοιχείων, τη στοιχειομετρία, τους αριθμούς k (rates) για την κάθε αντίδραση καθώς και τους δείκτες pointers που διαμορφώνουν τις αντιδράσεις. Με αυτά τα αρχεία που εξάγονται μέσω του SBML parser σε C αρχικοποιούνται οι μνήμες VT, RT, ST

και με ειδικό αλγόριθμο γραμμένο σε C και Python το Flag Table. Η μνήμη ST όπως φαίνεται και στο σχήμα 21 είναι διαχωρισμένη σε 2 μέρη ένα ολόκληρο και ένα χωρισμένο σε N τομείς. Αυτό γίνεται για να υποστηριχθούν και οι δύο τρόποι λειτουργίας δηλαδή για μία παράλληλη προσομοίωση αλλά και για N ξεχωριστές αντίστοιχα. Έτσι λοιπόν για μία N παράλληλη προσομοίωση χρησιμοποιείτε ολόκληρη η μνήμη ενώ για N ξεχωριστές διαφοροποιείτε σε τομείς. Όμως και στις δύο περιπτώσεις αρχικοποιούνται με τον ίδιο τρόπο δηλαδή μέσω των (SBML parsers) σε C και Python.

Αφού αρχικοποιηθούν οι μνήμες και γίνει η σύνθεση του κυκλώματος πρέπει να προγραμματιστεί η μονάδα ελέγχου Control Unit και πιο συγκεκριμένα ο καταχωρητής ελέγχου και αναλόγως να αρχίσει η διαδικασία. Ο χρήστης λοιπόν επιλέγει τι αλγόριθμο θα χρησιμοποιήσει στην προκειμένη περίπτωση τον FRM, τι τρόπο λειτουργίας θα έχει, εάν θα έχει μία παράλληλη προσομοίωση ή πολλές ανεξάρτητες, εάν θέλει στατιστικά στοιχεία, για πόσο χρόνο θέλει να διαρκέσει η προσομοίωση και επίσης ρυθμίζει εάν θέλει μία προσομοίωση (1 επανάληψη) μέχρι τον επιθυμητό χρόνο του αλγορίθμου ή πολλές προσομοιώσεις (multiple runs).

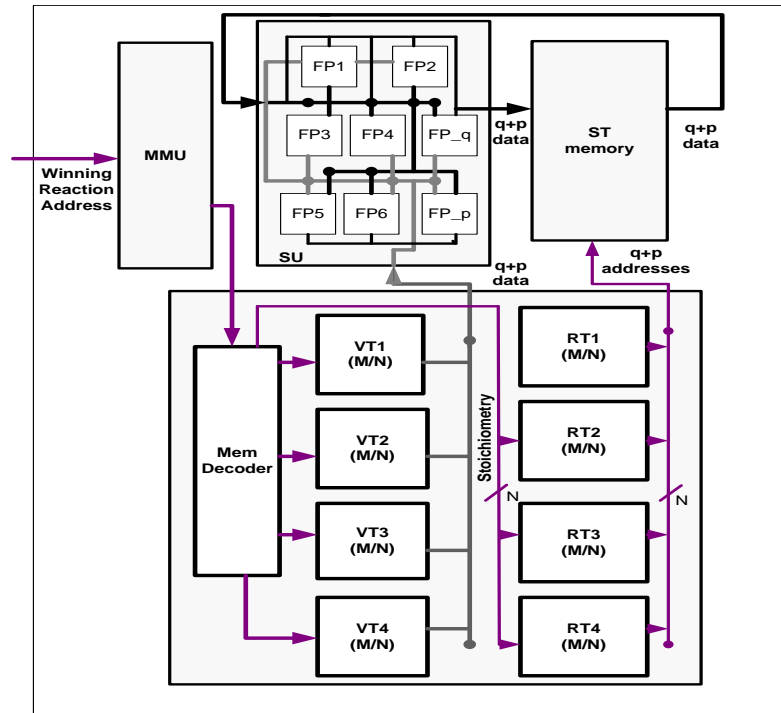
Μετά τον προγραμματισμό της μονάδος ελέγχου η ίδια η μονάδα ρυθμίζει όλες τις εσωτερικές υπό-μονάδες. Αρχικά λοιπόν ενεργοποιείτε η μονάδα διευθυνσιοδότησης (MMU) η οποία ρυθμίζει τις διευθύνσεις για όλες τις μνήμες ανάλογα με τον τρόπο λειτουργίας. Έτσι οι δείκτες στο Reaction Table δείχνουν ποια στοιχεία συμμετέχουν στην αντίδραση δηλαδή δείχνουν στη species Table (η οποία κρατά τις αρχικές συγκεντρώσεις) των στοιχείων που συμμετέχουν στην κάθε αντίδραση καθώς επίσης δίνουν και την σταθερά k (reaction rate) στα PEs. Με τη σειρά της η species Table δίνει με βάση τις διευθύνσεις που δέχεται από τους δείκτες (RT) τις συγκεντρώσεις στους επεξεργαστές (PEs) οι οποίοι υπολογίζουν την πιθανότητα αντίδρασης $\alpha_j(x)$ και τους χρόνους $\tau_j(x)$ με βάση την εξίσωση 21 του κεφαλαίου 2. Ακολούθως και σε στάδια πολλαπλής σωλήνωσης οι επεξεργαστές PE στέλνουν τα δεδομένα τους στα συστήματα εύρεσης ελαχίστου χρόνου (βλέπε μαύρη διαδρομή σχήμα 22). Αρχικά σειριακά από κάθε μονάδα εύρεσης αρχικού ελαχίστου χρόνου MTU1 βρίσκεται ο χρόνος αλλά και η διεύθυνση από την οποία προέκυψε αυτός ο ελάχιστος χρόνος η οποία αντανακλά στην διεύθυνση που δόθηκε από την MMU στους δείκτες pointers της RT μνήμης. Δηλαδή δείχνει την αντίδραση που οδήγησε στον ελάχιστο χρόνο. Έπειτα τα δεδομένα (ελάχιστος χρόνος και διεύθυνση που οδήγησε στον ελάχιστο χρόνο) από το 1^ο στάδιο συγκρίσεων για το κάθε επεξεργαστή δίδονται σε μια δενδροειδής δομή από συγκριτές την MTU2 από όπου θα βρεθεί ο τελικός ελάχιστος χρόνος και αντίστοιχα η διεύθυνση στην οποία αντιστοιχεί (μοβ διαδρομή σχήμα

22). Η διεύθυνση λοιπόν που αντιστοιχεί στον ελάχιστο χρόνο θα δοθεί πίσω στην MMU από όπου θα διαμοιραστούν οι διευθύνσεις απευθείας από την MMU στην Species Table για να δώσει τις συγκεντρώσεις στην μονάδα SU (Species Update) και επίσης η ίδια η διεύθυνση θα δοθεί και στην μονάδα VT (Stoichiometry Table) η οποία με τη σειρά της θα δώσει τη στοιχειομετρία της αντίστοιχα στην SU από όπου θα υπολογιστούν οι νέες συγκεντρώσεις (σχήμα 23).



Σχήμα 22: Διαδρομή δεδομένων στο FRM4X SoC για τον υπολογισμό της επόμενης αντίδρασης.

Παράλληλα θα έχει υπολογιστεί από την μονάδα TU ο νέος χρόνος αντίδρασης $t = t_{old} + \tau_{min}$. Άρα παρατηρούμαι ότι δημιουργείται ένα στιγμιότυπο εξέλιξης του φαινομένου. Οι νέες συγκεντρώσεις θα καταγραφούν πίσω στην Species Table και η διαδικασία θα συνεχιστεί μέχρις ότου ο χρόνος γίνει ίσος με κάποια τιμή, δηλαδή όταν $t = t_{sim}$ τον οποίο χρόνο t_{sim} έχει θέσει ο χρήστης σαν μεταβλητή. Αυτή ήταν η περιγραφή της διαδικασίας μια παράλληλης προσομοίωσης για τον αλγόριθμο FRM.



Σχήμα 23: Διαδρομή δεδομένων στο FRM4X SoC για τον υπολογισμό των συγκεντρώσεων των στοιχείων αντίδρασης.

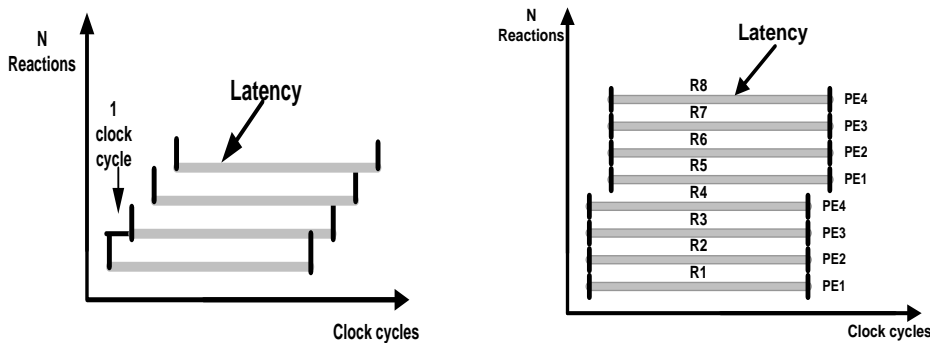
Αξίζει να σημειωθεί ότι για τις παράλληλες προσομοιώσεις και ανάλογα με τον αριθμό των επεξεργαστών αλλάζει και ο διάυλος μεταφοράς δεδομένων, ένας ακόμη λόγος για την ύπαρξη αναδομήσιμης (reconfigurable) παραμετροποιημένης αρχιτεκτονικής.

Η MMU θα πρέπει να δώσει N διευθύνσεις στα N Reaction Tables από όπου $N^*(q)$ διευθύνσεις θα πρέπει να δοθούν στις μνήμες (ST) species Table για όλη την διαδικασία μέχρι να βρεθεί ο ελάχιστος χρόνος τ_{jmin} και να υπολογιστούν και οι νέες διευθύνσεις. Οι μεταβλητές q και p εκφράζουν αντίστοιχα το βαθμό αντίδρασης δηλαδή τον αριθμό των αντιδρώντων που είναι q (reaction order) και p είναι ο αριθμός των προϊόντων που παράγονται. Επίσης αντίστοιχα υπάρχουν N^*VT μνήμες, $N^*(q+p)$ SU μονάδες που χρησιμοποιούνται και για τους δύο τρόπους λειτουργίας δηλαδή για μία παράλληλη προσομοίωση και N παράλληλες ανεξάρτητες. Επίσης το ίδιο ισχύει και για τη μονάδα TU(time update).

Η μνήμη (ST) Species Table που κρατά τις συγκεντρώσεις είναι χωρισμένη στα δύο όπως αναφέρθηκε και προηγουμένως για τους δύο τρόπους λειτουργίας. Δηλαδή για μία N παράλληλη προσομοίωση στέλνονται δεδομένα από τις $N(q+p)$ εξόδους της ST. Ο διάυλος δεδομένων που μεταφέρει τις συγκεντρώσεις των δεδομένων από τις ST μνήμες στους PE επεξεργαστές έχει μέγεθος $N^*(q) * 32$ bits. Επίσης όσο αναφορά το Flag Table που φέρει σήματα για τον προγραμματισμό όπως θα δούμε της μονάδος υπεύθυνης για τον υπολογισμό της πιθανότητας αντίδρασης, έχει ένα πίνακα (array) μεταβλητό όπως και όλες οι μνήμες που εξαρτάται από τον αριθμό των αντιδράσεων M , των $(q+1)$ bits όπου q ο

αριθμός της τάξης της αντίδρασης (αριθμός αντιδρώντων). Επίσης δέχεται N διευθυνσιοδοτήσεις από την MMU και δίνει $N \cdot q$ δεδομένα στην είσοδο των επεξεργαστών PE.

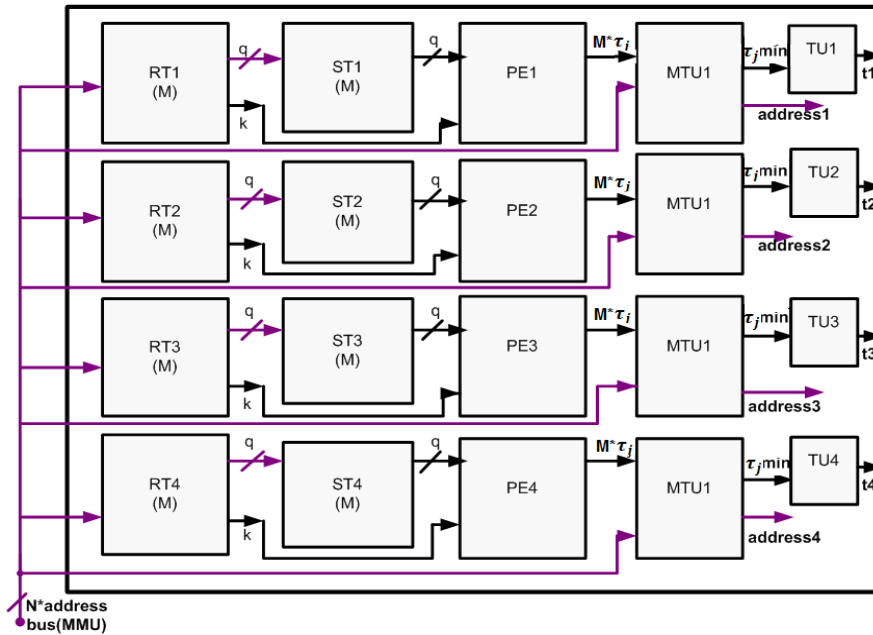
Οι επεξεργαστές PEs με τη σειρά τους θα στείλουν δεδομένα των 32 bits οπότε ο δίαυλος μεταφοράς δεδομένων προς τις μονάδες MTUs είναι $N \cdot 32$. Επίσης ανάλογα με τον αριθμό των αντιδράσεων και τον αριθμό των στοιχείων προς αντίδραση ρυθμίζεται και το εύρος των μνημών (array). Για το συγκεκριμένο FPGA στο οποίο ελέγχθηκε το FRM4X SoC και για $N=4$ επεξεργαστές το μέγιστο εύρος του πίνακα μνήμης (array) είναι 4096 που σημαίνει ότι μπορούν να προσομοιωθούν $4 \cdot 4096$ αντιδράσεις. Με βάση την προηγούμενη ανάλυση εύλογα παρατηρούμαι ότι μιλάμε για μία παραμετροποιημένη αρχιτεκτονική.



Σχήμα 24: Απεικόνιση εκτέλεσης της FRM προσομοίωσης με $N=1$ επεξεργαστή και $N=4$ παράλληλους.

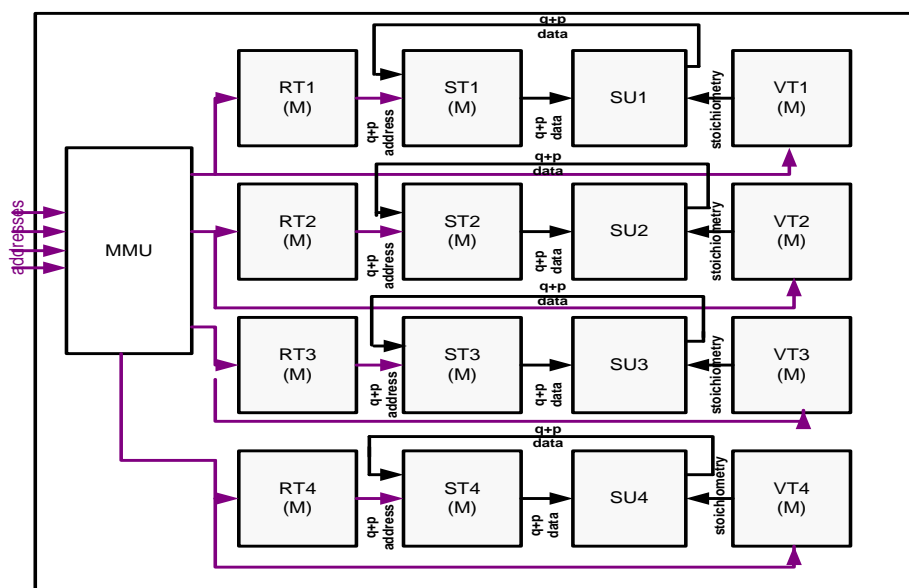
Για να εκτελεστούν N ανεξάρτητες προσομοιώσεις θα πρέπει να προγραμματιστεί η μονάδα ελέγχου σε τρόπο λειτουργίας ανεξάρτητων προσομοιώσεων και ακολούθως η μονάδα ελέγχου να ενεργοποιήσει και να προγραμματίσει ανάλογα όλες τις επιμέρους μονάδες. Σε περίπτωση που έχω διαφορετικά βιοχημικά μοντέλα για προσομοίωση θα αρχικοποιηθούν οι μνήμες για το κάθε μοντέλο μέσω του προγράμματος SysPy. Έτσι θα έχω 4 διαφορετικές αρχικοποιήσεις για τα RT,VT,ST, και για τα Flag Tables (FG).

Επίσης αλλάζει και ο τρόπος διευθυνσιοδότησης από την MMU. Τώρα στην περίπτωση του FRM4X είναι σαν να εκτελούνται 4 διαφορετικοί FRM1X ταυτόχρονα. Η MMU θα δώσει N διευθύνσεις στη μνήμη δεικτών (pointers) RT και Flag Tables. Οι δείκτες με τη σειρά τους θα δώσουν τις διευθύνσεις, θα δείξουν δηλαδή στη χωρισμένη μνήμη ST σε τομείς όπου από εκεί η ST θα δώσει σε κάθε PE τα δεδομένα. Το κάθε PE θα υπολογίσει ακολούθως το χρόνο $\tau_j(x)$ και $\alpha_j(x)$ για όλες τις αντιδράσεις του μοντέλου. Αντίστοιχα θα βρεθεί ο ελάχιστος χρόνος από τη μονάδα MTU1 όπου τώρα δεν θα ενεργοποιηθούν οι μονάδες MTU2 αφού έχω N διαφορετικά $\tau_j(x)$ και $\alpha_j(x)$ οπότε η διεύθυνση που αντιστοιχεί στο μικρότερο χρόνο για την κάθε προσομοίωση θα δοθεί πίσω στην MMU. Συνολικά η MMU θα δώσει 4 διαφορετικές διευθύνσεις που αντιστοιχούν σε τέσσερις ελάχιστους χρόνους.



Σχήμα 25: Διαδρομή δεδομένων στο FRM4X SoC για τον υπολογισμό των N επόμενων αντιδράσεων σε τρόπο λειτουργίας 2'.

Άρα η ST θα πρέπει να πάρει 4 διαφορετικές διευθύνσεις απευθείας από την MMU για $N=4$ και για τον κάθε τομέα θα γίνουν 4 διαφορετικοί υπολογισμοί για να υπολογιστούν οι νέες συγκεντρώσεις από τις $4*(q+p)$ Species Update (SU) (βλέπε σχήμα 26) μονάδες, όπως επίσης και για να υπολογιστούν 4 διαφορετικά στιγμιότυπα εξέλιξης του χρόνου $4*(t = t_{old} + \tau_{min})$ από την μονάδα TU μέχρι ο κάθε χρόνος προσομοίωσης να φτάσει την τιμή που έχει οριστεί για την κάθε προσομοίωση από τον χρήστη.



Σχήμα 26: Διαδρομή δεδομένων στο FRM4X SoC στον 2^ο τρόπο λειτουργίας για τον υπολογισμό N διαφορετικών συγκεντρώσεων των στοιχείων των N διαφορετικών βιομοντέλων.

Σημαντικό επίσης στοιχείο είναι ότι και στους δύο τρόπους υπάρχει η δυνατότητα πολλαπλών προσομοιώσεων δηλαδή μόλις το $t = t_{max}$ μπορεί η διαδικασία να ξαναρχίσει από την αρχή. Εσωτερικά υπάρχουν κρυφά ST αντίγραφα της μνήμης με τις αρχικές συγκεντρώσεις και για τους δύο τρόπους λειτουργίας οπότε μετά το πέρας μιας προσομοίωσης (1 επανάληψη) στέλνονται οι αρχικές συγκεντρώσεις στην ST προς επεξεργασία.

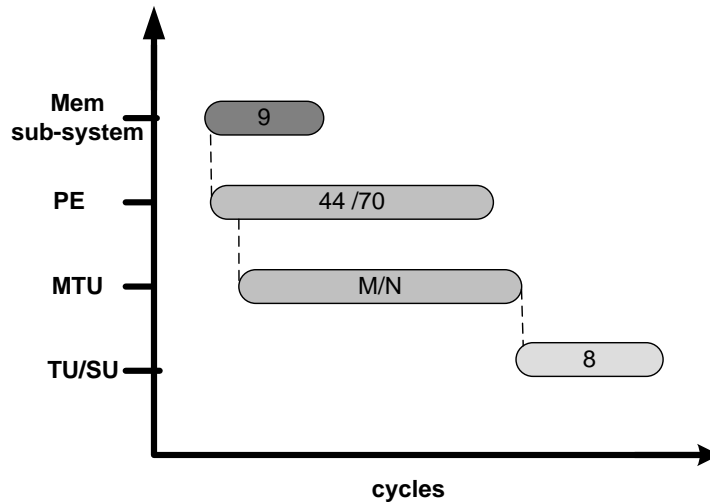
Μία άλλη λειτουργία της μονάδας είναι ότι και στις 2 μεθόδους υπάρχουν υπό τρόποι λειτουργίας. Δηλαδή μπορεί ο χρήστης προγραμματίζοντας τη μονάδα να ζητήσει να εξελίσσεται η αντίδραση μέχρι εξαντλήσεως των στοιχείων δηλαδή μέχρι οι καινούργιες συγκεντρώσεις να γίνουν όλες '0' που σημαίνει ότι η προσομοίωση δεν θα σταματήσει μέχρι $t = t_{max}$ αλλά μέχρι κάποια ή όλες οι συγκεντρώσεις να γίνουν '0' όπου και σε αυτόν τον υπό-τρόπο λειτουργίας ισχύει η έννοια των πολλαπλών προσομοιώσεων.

Επίσης εάν στο 2^ο τρόπο λειτουργίας τα βιομοντέλα προς προσομοίωση είναι μικρότερα από τον αριθμό των επεξεργαστών τότε μπορεί ο χρήστης ανάλογα με την πολυπλοκότητα να επιλέξει ποιο μοντέλο θέλει να παραλληλοποιηθεί. Για παράδειγμα εάν έχω 2 μοντέλα στο NRM_FRM4X εκμεταλλευόμενοι την παραλληλοποίηση της αρχιτεκτονικής θα έχουμε αντίστοιχα 2* NRM2X ή FRM2X προσομοιώσεις.

Η δυνατότητα πολλαπλών προσομοιώσεων κρίθηκε αναγκαία για να υπάρξει καλύτερη σύγκλιση των τιμών των συγκεντρώσεων των χημικών στοιχείων της στοχαστικής προσομοίωσης γύρω από την πραγματική τιμή και αυτό γιατί όπως αναλύσαμε και στο κεφάλαιο 2 υπάρχει η έννοια του τυχαίου (random) στην στοχαστική προσομοίωση και έτσι παρατηρείται στην κάθε επανάληψη του αλγορίθμου (multiple iterations) μικρές διαφοροποιήσεις ως προς τις συγκεντρώσεις. Για το λόγο αυτό συγκεντρώνονται από την κάθε διαφορετική προσομοίωση, οι εκάστωτε συγκεντρώσεις και υπολογίζεται ο μέσος όρος για την συγκέντρωση του κάθε στοιχείου της βιοχημικής αντίδρασης και για τους δύο αλγορίθμους και στους δύο τρόπους λειτουργίας.

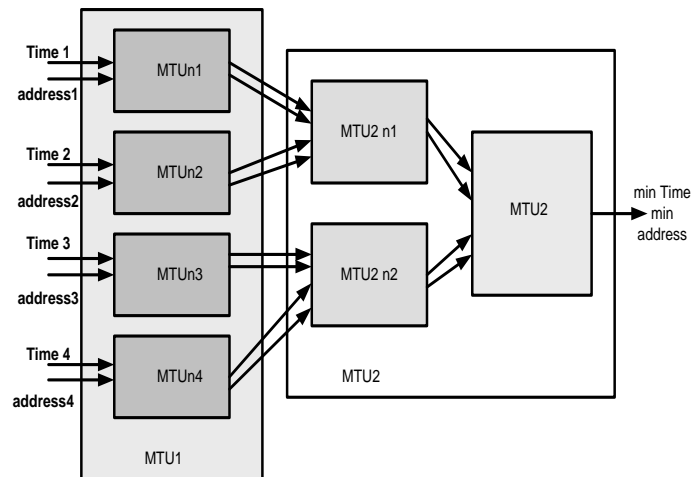
3.4 Ανάλυση μίας N παράλληλης προσομοίωσης

Αναλυτικότερα σε αυτό το κεφάλαιο θα αναλύσουμε το πρώτο τρόπο λειτουργίας και τις υπό μονάδες καθώς και το πώς αυτές συνεργάζονται για να επιτύχουν την προσομοίωση. Το παρακάτω διάγραμμα δείχνει τον καταμερισμό των λειτουργιών σε νήματα (threads) και πώς αυτά κατανέμονται στις βασικές μονάδες. Επίσης μπορούμε να παρατηρήσουμε και τους κύκλους καθυστέρησης (latency) (σχήμα 27).



Σχήμα 27: Διάγραμμα κατανομής των εργασιών (threads) για την εκτέλεση του FRM.

Επιπλέον θα δείξουμε αναλυτικά την παραμετροποίηση των βασικών υπό μονάδων ανάλογα με τον αριθμό παραλληλοποίησης N και του αριθμού των βιοχημικών αντιδράσεων M που καλείται το SoC να επεξεργαστεί. Αρχικά η μονάδα ελέγχου ενημερώνεται όπως είπαμε για τον τρόπο λειτουργίας. Με της σειρά της στέλνει τα κατάλληλα σήματα στην MMU για τον τρόπο διευθυνσιόδοτησης. Στο σημείο αυτό πρέπει να αναφέρουμε όπως έγινε κατανοητό από το προηγούμενο κεφάλαιο ότι ο αριθμός των PEs θα καθορίσει και τον αριθμό των μνημών ST,RT,VT και Flag Tables καθώς και την δενδροειδής δομή των (MTU2) δηλαδή εάν θα έχω $\log_2(N)$ δομές (σχήμα 28).



Σχήμα 28: Διάγραμμα δενδροειδούς δομής των MTUS.

Επίσης θα έχω $N*TU$ μονάδες για τον υπολογισμό του στιγμιότυπου χρόνου αντίδρασης και $N*(q+p)$ SU για τον υπολογισμό των συγκεντρώσεων από όπου για τον συγκεκριμένο τρόπο προσομοίωσης θα χρησιμοποιηθούν $(q+p)$ μονάδες όπου q είναι η τάξη της αντίδρασης και p ο αριθμός των προϊόντων. Η MMU μπορεί να χειριστεί και τους 2 τρόπους λειτουργίας και για αυτό φέρει τις κατάλληλες εξόδους–εισόδους και για τους 2

τρόπους. Άρα ουσιαστικά θα δίνει πάντα στις $N*RT$ μνήμες N διευθύνσεις. Στο συγκεκριμένο τρόπο προσομοίωσης το SysPy που δημιουργεί τον HDL κώδικα για όλες τις μνήμες με βάση το SBML μοντέλο θα πρέπει να διασπάσει το μοντέλο διά του αριθμού των επεξεργαστών στις μνήμες ώστε να μπορεί να γίνει μία N παράλληλη προσομοίωση. Το ίδιο θα πρέπει να γίνει και στην μνήμη VT. Επίσης από την διάσπαση των αντιδράσεων θα δημιουργηθούν και θα κατανεμηθούν με βάση έναν αλγόριθμο που έχει αναπτυχθεί σε SysPy και η μνήμη των Flags. Άρα εάν έχω ένα μοντέλο 512 αντιδράσεων θα έχω $512/4 = 128$ αντιδράσεις που θα πρέπει να εκτελεστούν για το κάθε PE, το οποίο μου μειώνει τις εσωτερικές καθυστερήσεις και αυξάνει την απόδοση του συστήματος για μεγάλα μοντέλα. Οι RT μνήμες που φέρουν τις αντιδράσεις δέχονται N διευθύνσεις και δίνουν στην περίπτωση του συγκεκριμένου τρόπου λειτουργίας $N*(q+p)$ διευθύνσεις (δεδομένα) στην ST μνήμη που στην περίπτωση του 1^{ου} τρόπου λειτουργίας χρησιμοποιείται όλος ο πίνακας (array) αυτούσιος. Από τις $N*(q+p)$ διευθύνσεις θα χρησιμεύσουν $N* q$ σε 1^η φάση για τον υπολογισμό των χρόνων τ_j από τα στοιχεία της ST μνήμης και $(q+p)$ διευθύνσεις για τον υπολογισμό των συγκεντρώσεων της αντίδρασης που κέρδισε και αυτό γιατί αυτή η αντίδραση θα αποτελείται από q αντιδρώντα και p προϊόντα. Άρα η ST μνήμη για το ένα της κομμάτι θα έχει $N*(q+p)$ εξόδους δεδομένων. Από αυτές τις εξόδους $N*q$ πηγαίνουν στα PEs μαζί με τον αριθμό k (reaction rate) το οποίο έρχεται από την RT μνήμη και είναι χαρακτηριστική σταθερά της κάθε αντίδρασης για να υπολογισθούν τα $\tau_j(x)$ και $a_j(x)$. Για τον υπολογισμό των αντιδράσεων από την μονάδα SU θα χρειαστούμε τις $(q+p)$ εξόδους της ST καθώς επίσης και μία εκ των $(q+p)$ εξόδων από τις $N*VT$ μνήμες εκ των οποίων η VT μνήμη φέρει την στοιχειομετρία της αντίδρασης που κέρδισε και η οποία θα δώσει στις $(q+p)$ εξόδους της την κατάλληλη στοιχειομετρία για να υπολογισθούν οι νέες συγκεντρώσεις. Οι συγκεντρώσεις αυτές θα ξαναγραφούν απευθείας μέσω της MMU στην ST από την είσοδο εγράφης όπου θα χρειαστούν $(q+p)$ κύκλοι εγράφης. Οι θύρες εγράφης των δεδομένων είναι χαρακτηριστικές των Block Rams για την χρησιμοποίηση ενός πίνακα (array). Γενικά ο αριθμός των θυρών εισόδου για την εγγραφή εξαρτάται κατά αναλογία από τον αριθμό των πινάκων δεδομένων (arrays) που χρησιμοποιούνται από μία περιγραφή συμπεριφοράς της μνήμης Block Ram σε VHDL και αυτό γίνεται για την καλύτερη σύνθεση των μνημών από τον XST της Xilinx ή το Quatrus Altera.

Σε αυτό το σημείο αξίζει να σημειωθεί ότι χρησιμοποιείτε ένας ολόκληρος μηχανισμός από πολυπλέκτες οι οποίοι συνεργάζονται με τις μονάδες MTU1 και MTU2 για να βρεθεί η διεύθυνση εκείνη που αντιστοιχεί στον ελάχιστο χρόνο και για να επιλεγούν τα δεδομένα που αντιστοιχούν στη διεύθυνση που έχει προκύψει από τις $N*RT$ μνήμες και η οποία αντιστοιχεί στην αντίδραση που θα εκτελεστεί στο επόμενο στιγμιότυπο χρόνου (σχήμα 23

Mem Decoder). Τα MTU2 μαζί με σήματα (flags) που έρχονται από τους εσωτερικούς καταχωρητές που φέρουν τα δεδομένα των μνημών RT υπολογίζουν ποια αντίδραση κέρδισε και από ποιο RT προήλθε.

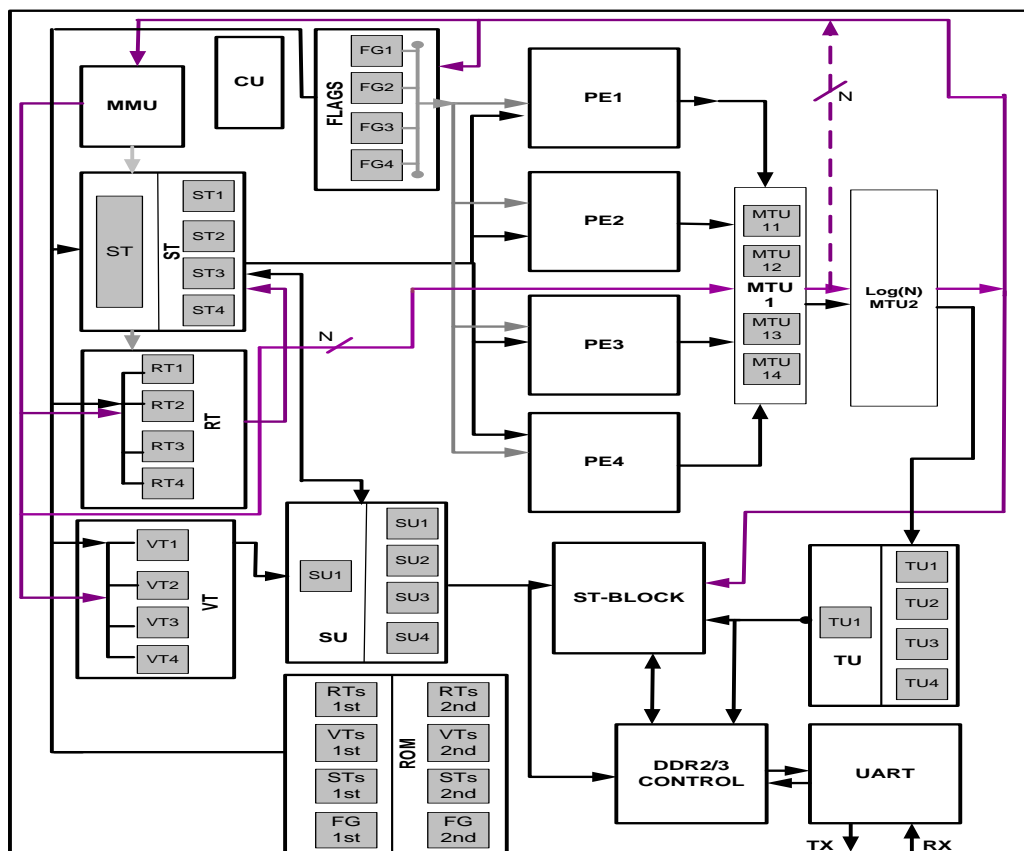
Η MMU λοιπόν με βάση την διεύθυνση που υποδηλώνει την αντίδραση που κέρδισε θα δώσει αυτή την διεύθυνση μαζί με μία χαρακτηριστική ετικέτα (flag). Η ετικέτα αυτή είναι χαρακτηριστική του κάθε εκ των N RT. Η ετικέτα (flag) εισέρχεται σε ένα σύστημα αποκωδικοποιητών για να ενεργοποιηθεί η κατάλληλη RT μνήμη που θα δώσει την αντίδραση που θα πραγματοποιηθεί στο επόμενο Δt . Από εκεί οι κατάλληλες διευθύνσεις από την RT θα δοθούν στην ST για να δώσει τα στοιχεία των συγκεντρώσεων προς υπολογισμό στην μονάδα SU (Species Update). Επίσης θα γίνει και η ενημέρωση του χρόνου από μία εκ των N μονάδων (Time Update Units) που χρησιμοποιούνται σε αυτόν τον τρόπο λειτουργίας. Οι συγκεντρώσεις των στοιχείων της βιοχημικής αντίδρασης που πραγματοποιήθηκε αφού ενημερωθούν και αλλάξουν οι τιμές τους από την μονάδα SU θα εγγραφούν στην μνήμη ST σε $(q+p)$ κύκλους.

3.5 Περιγραφή N παράλληλων προσομοιώσεων

Στην παράγραφο αυτή θα αναλύσουμε με πιο λεπτομέρεια την λειτουργία N παράλληλων προσομοιώσεων. Όπως παρατηρούμε και από το σχήμα 29 οι διαδρομές με ανοιχτό γκρι εκφράζουν την λειτουργία των N παράλληλων προσομοιώσεων. Αρχικά μέσω του προγράμματος SysPy φορτώνονται τα επιμέρους xml μοντέλα και αρχικοποιούνται όλες οι μνήμες ανάλογα με το μοντέλο. Δηλαδή θα έχω N διαφορετικές αρχικοποιήσεις για την RT και για την ST και πιο συγκεκριμένα θα χρησιμοποιηθεί το 2^ο κομμάτι της ST, το οποίο είναι χωρισμένο σε N υπό κομμάτια ανεξάρτητα μεταξύ τους. Επίσης για την VT μνήμη θα έχω εξίσου N διαφορετικές αρχικοποιήσεις καθώς και για το Flag Table το οποίο στα N τμήματα του θα έχει διαφορετική αρχικοποίηση που εξαρτάται από το εκάστωτε SBML μοντέλο προς προσομοίωση. Η μονάδα ελέγχου θα προγραμματιστεί σε αυτόν τον τρόπο λειτουργίας και πιο συγκεκριμένα θα δώσει τα κατάλληλα σήματα στην MMU όπου πάλι θα δοθούν N διαφορετικές διευθύνσεις στις RT και VT μνήμες, και στα Flag Tables όπου οι RT μνήμες θα δώσουν N διευθύνσεις στο 2^ο κομμάτι της ST το χωρισμένο διά N . Γενικά οι μνήμες σε αυτόν τον τρόπο λειτουργίας μπορεί να περιέχουν ανεξάρτητες διευθύνσεις και δεδομένα για να πραγματοποιηθούν οι ανεξάρτητες προσομοιώσεις.

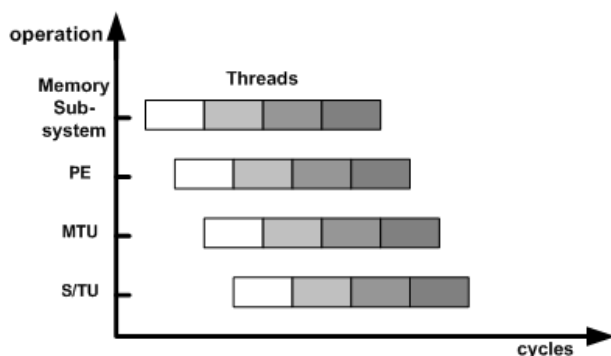
Αρχικά λοιπόν για το κάθε PE θα δοθούν από την ST $N \cdot q$ δεδομένα από το κάθε κομμάτι της ST στα PEs από όπου θα υπολογιστούν (μαζί με τα k (reaction rates) που θα δοθούν επίσης από τις RT μνήμες), N διαφορετικοί χρόνοι όπου οι τιμές των χρόνων αυτών θα μεταβούν για να βρεθεί ο ελάχιστος χρόνος για την κάθε προσομοίωση στις N

σειριακές βαθμίδες MTUs της βαθμίδος MTU1. Από την MTU1 που περιέχει N διαφορετικούς σειριακούς συγκριτές, οι N διαφορετικές διευθύνσεις που αντιστοιχούν στις αντιδράσεις που θα γίνουν για την κάθε προσομοίωση θα οδηγηθούν απευθείας προς την MMU από όπου θα δοθούν N διαφορετικές διευθύνσεις στις RT και VT μνήμες για να γίνουν N διαφορετικοί υπολογισμοί των συγκεντρώσεων των στοιχείων για τις διαφορετικές αυτές αντιδράσεις. Η μονάδα SU υπεύθυνη για τον υπολογισμό των συγκεντρώσεων θα αποτελείται από $N*(q+p)$ αθροιστές (IEEE 754 floating point units) και θα υπολογίσει $N*(q+p)$ συγκεντρώσεις από όπου για το κάθε κομμάτι του ST θα γίνουν παράλληλα $(q+p)$ εγγραφές των συγκεντρώσεων. Επίσης παράλληλα θα έχουν γίνει και N υπολογισμοί του στιγμιότυπου του χρόνου από τις $N*Tu$ s βαθμίδες.



Σχήμα 29: Διάγραμμα Αρχιτεκτονικής FRM με $N=4$ επεξεργαστές για 1 και N παράλληλες προσομοιώσεις.

Παρακάτω φαίνεται ο διαμοιρασμός των N διαφορετικών αντιδράσεων προς υπολογισμό σε νήματα (threads) στις βασικές υπό- μονάδες.

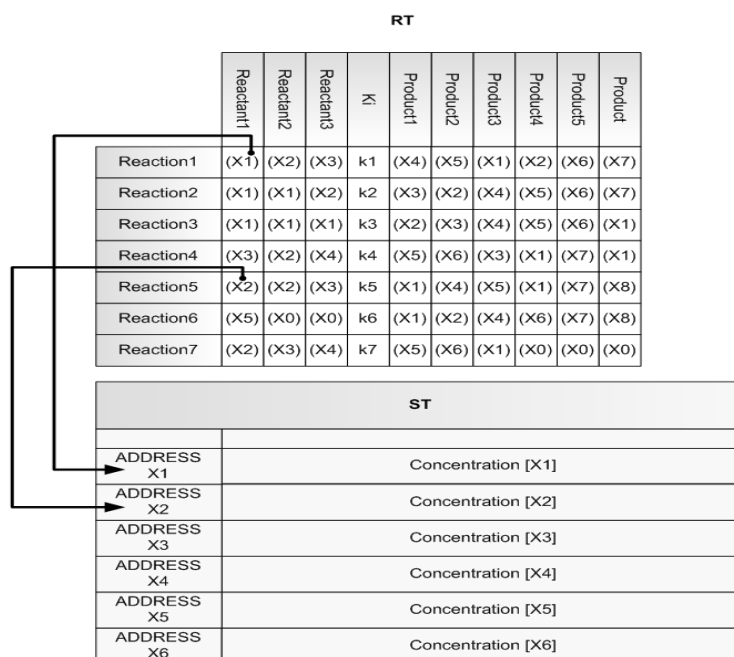


Σχήμα 30: Διάγραμμα με παράλληλα νήματα (threads) στην κάθε λειτουργική υπόμονάδα για την εκτέλεση N παράλληλων FRM.

Στο σημείο αυτό πρέπει να σημειωθεί ότι υπάρχουν ειδικοί μηχανισμοί για τον συγχρονισμό της λειτουργίας των επιμέρους αντιδράσεων. Εάν κάποιο μοντέλο απαιτεί παραπάνω χρόνο προσομοίωσης τότε πριν ξαναρχίσει από την αρχή μια καινούργια προσομοίωση το σύστημα περιμένει το συγκεκριμένο μοντέλο να τελειώσει όλη την προσομοίωση του. Επίσης πριν ξεκινήσει από την αρχή η διαδικασία κρατιέται ακριβές αντίγραφο με τις αρχικές συγκεντρώσεις για το 1^ο ή το 2^ο κομμάτι της μνήμης ST ανάλογα με τον τρόπο λειτουργίας. Με αυτόν τον τρόπο προσομοίωσης μπορούν να προσομοιωθούν και να συγκριθούν ενδεχομένως παράλληλα αλλά και γρήγορα διαφορετικά βιοχημικά μοντέλα. Επιπλέον η σχεδίαση επιτρέπει και στατιστική επεξεργασία για το κάθε βιοχημικό μοντέλο ή και συγκρίσεις μεταξύ αυτών από την μονάδα STU (St-block). Πέραν της στατιστικής επεξεργασίας για την εύρεση των μέσων τιμών των συγκεντρώσεων από όλες τις προσομοιώσεις σε κάθε χρονική στιγμή κρίθηκε αναγκαία η διαδικασία αυτή και στην περίπτωση που για το ίδιο βιοχημικό μοντέλο θέλαμε μια μελέτη για την μεταβολή των στοιχείων για διαφορετικά (reaction rates k). Όπου μετά το πέρας μίας προσομοίωσης (1 run) για το επιλεχθέν στοιχείο ο χρήστης της πλατφόρμας θα μπορούσε να εξάγει τον μέσο όρο και την τυπική απόκλιση για το κάθε στοιχείο.

3.6 Οργάνωση μνήμης

Σε αυτό το κεφάλαιο θα περιγράψουμε την οργάνωση των μνημών με πιο λεπτομέρεια αναλύοντας τις ιδιαιτερότητες της αρχιτεκτονικής για τον κάθε τρόπο λειτουργίας. Παρακάτω φαίνεται η λειτουργία για τον 1^ο τρόπο λειτουργίας και συγκεκριμένα μπορούμε να δούμε πως γίνεται η αντιστοίχιση μεταξύ της RT μνήμης που περιέχει τους δείκτες και της ST μνήμης που περιέχει τα στοιχεία.

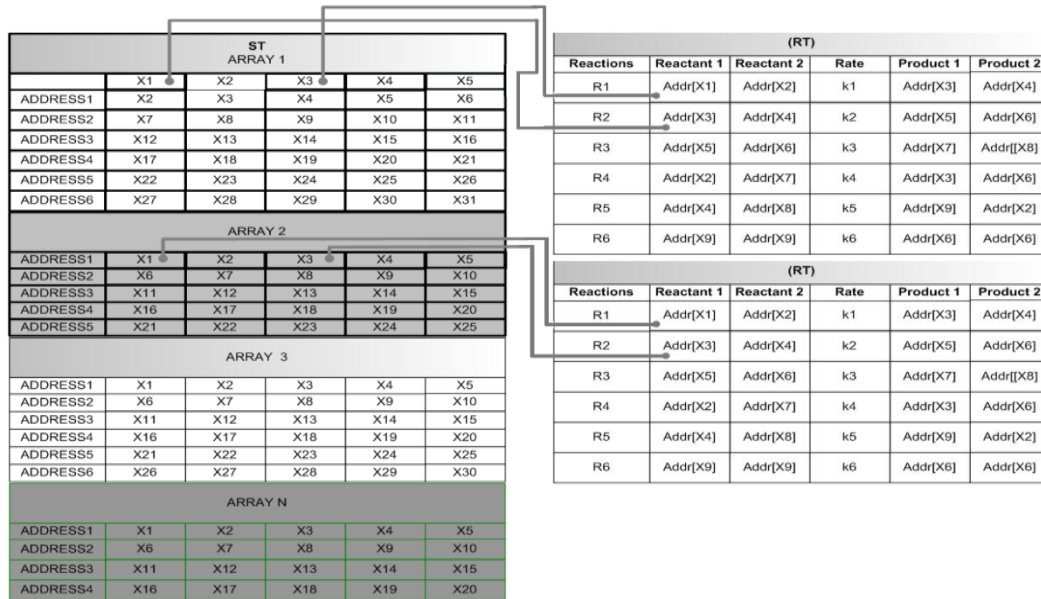


Σχήμα 31: Οργάνωση μνημών (Species Table και Reaction table για μια παράλληλη προσομοίωση)

Όπως παρατηρούμαι η RT μνήμη περιέχει τις αντιδράσεις. Δηλαδή μια αντίδραση 2^{ου} βαθμού της μορφής R1: $X_1 + X_2 = k > X_3 + X_4$ θα περιγράφεται από την RT μνήμη για την 1^η διεύθυνση του πίνακα ως εξής:

Θα υπάρχουν 4 pointers και ο αριθμός k (reaction rates) στον πίνακα από bits της 1^{ης} διεύθυνσης αφού μιλάμε για την 1^η αντίδραση. Δηλαδή ο χώρος που καταλαμβάνει η πληροφορία για την κάθε αντίδραση για την 1^{ης} θέση στον πίνακα θα έχει τη μορφή : $X''00000001000000023f8000000000000300000004''$ όπου οι δείκτες (pointers) δίνονται σε hex μορφή όπου έστω το $k= '1'$ και σε IEEE 754 float hex είναι το $''3f800000''$. Η σταθερά k θα περάσει αυτούσια στα PEs για αυτό και είναι σε IEEE 754 hex μορφή.

Το σχήμα 32 δείχνει ακριβώς το προηγούμενο παράδειγμα αλλά για αντιδράσεις μέχρι 4^{ου} βαθμού δηλαδή 4 αντιδρώντα. Ο ίδιος τρόπος οργάνωσης ισχύει και για την μνήμη VT χωρίς την καταχώρηση του αριθμού k δηλαδή ο πίνακας αποτελείται από bits με στοιχεία των 8 δεκαεξαδικών ψηφίων. Αυτός ο τρόπος οργάνωσης ισχύει για όλες τις N RT και VT μνήμες. Η ST μνήμη περιέχει σε κάθε διεύθυνση του πίνακα ή των πινάκων εάν είναι N χωρισμένη τον αριθμό της συγκέντρωσης σε IEEE hex 754. Ο αριθμός αυτός περιγράφει πόσα στοιχεία υπάρχουν για τη συγκεκριμένη συγκέντρωση εκφρασμένη σε mol/l. Το παρακάτω σχήμα δείχνει την οργάνωση και την συσχέτιση πάλι της RT με την ST μνήμη για N ανεξάρτητες διαφορετικές περιπτώσεις.



Σχήμα 32: Οργάνωση μηνυών (Species Table και Reaction table για N παράλληλες προσομοιώσεις)

Η διαδικασία είναι ακριβώς η ίδια με πριν για τον 1^ο τρόπο λειτουργίας και οι αρχικοποιήσεις των μηνυών και οι αντιστοιχίσεις ακριβώς οι ίδιες. Το μόνο που αλλάζει είναι ο διαχωρισμός του ST σε κομμάτια, οπότε η κάθε RT, VT μνήμη αντιστοιχεί τις διευθύνσεις και αντίστοιχα τα δεδομένα της στο κάθε κομμάτι από τα N της ST.

Αξίζει να σημειωθεί ότι το εύρος του πίνακα (array) για όλες της μνήμες έχει μέγεθος που εξαρτάται από τον αριθμό των αντιδράσεων M και αντίστοιχα για την ST από τον αριθμό των στοιχείων S .

Ο πίνακας Flag Table φέρει την πληροφορία που ενεργοποιεί τη σωστή διαδικασία για τον υπολογισμό της πιθανότητας αντίδρασης $\alpha_j(x)$ στην μονάδα PE με βάση το βαθμό της αντίδρασης (reaction order) που φαίνεται στον πίνακα 1 του κεφαλαίου 2. Αυτή η μνήμη δέχεται τις διευθύνσεις από την MMU για την κάθε αντίδραση και η κάθε διεύθυνση στον πίνακα αποτελείται από $q+1$ τον αριθμό bits όπου q είναι ο αριθμός των αντιδρώντων που συμμετέχουν σε μία αντίδραση.

3.7 Μονάδα PE

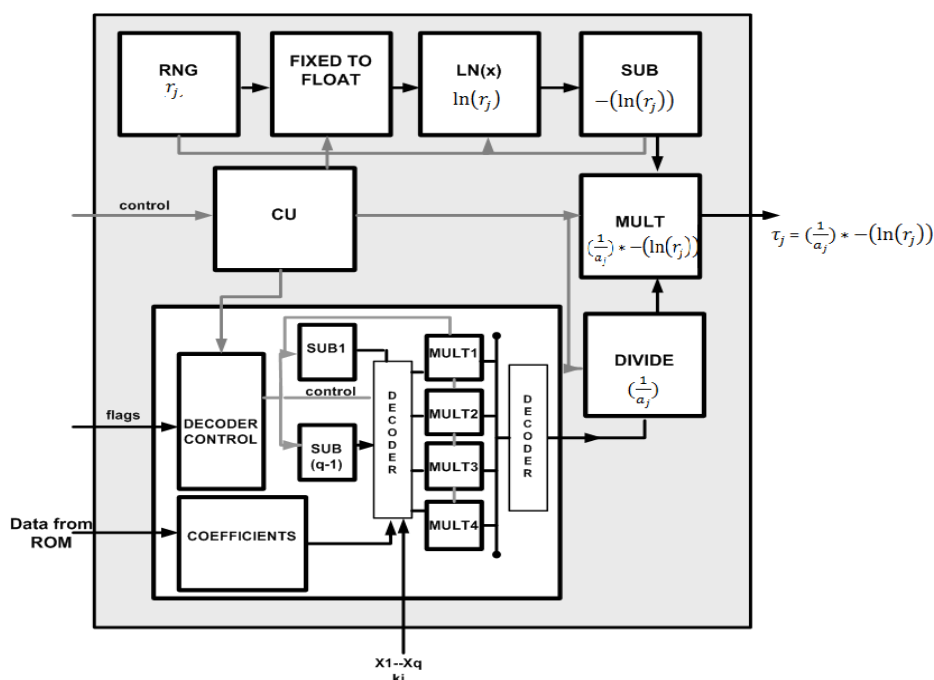
Η μονάδα αυτή χαρακτηρίζεται ως η πιο σημαντική στην σχεδίαση αυτή καθώς φέρει το κύριο επεξεργαστικό βάρος και καθορίζει και την συχνότητα λειτουργίας καθώς αποτελεί το κρίσιμο μονοπάτι της σχεδίασης. Η μονάδα αποτελείται από IEEE 754 floating point βαθμίδες μονής ακρίβειας (single precision). Το σχήμα (33) δείχνει όλες τις υπό μονάδες της μονάδος PE. Πριν ξεκινήσουμε την ανάλυση παρατηρούμαι ότι η εξίσωση που υπολογίζει το $\tau_j(x)$ μπορεί να απλοποιηθεί περεταίρω στην εξίσωση (25)

$$\tau_j(x) = \frac{1}{a_j} * - \ln(r_j) \quad (25)$$

Αυτό το βήμα για τον υπολογισμό του χρόνου μειώνει αρκετά την επεξεργαστική ισχύ της μονάδος. Με βάση την εξίσωση (25) θα αναλύσουμε την μονάδα αυτή.

Χωρίζουμε την εξίσωση 25 σε 2 μέρη, το 1^ο μέρος υπολογίζεται στο κάτω μέρος της μονάδας PE και το 2^ο μέρος στο επάνω μέρος όπου τα δεδομένα ακολουθούν μια διαδικασία σωλήνωσης έχοντας όλες τις υπό-μονάδες σωληνωμένες εσωτερικά. Επειδή απαιτείται συγχρονισμός όλων των επιμέρους μονάδων υπάρχει μία μονάδα ελέγχου η οποία συνεχώς ανταλλάσει μηνύματα μέσω σημάτων με την κύρια μονάδα ελέγχου όλου του συστήματος. Αρχικά αναλύοντας το πάνω μονοπάτι παρατηρούμαι την γεννήτρια παραγωγής ψευδοτυχαίων αριθμών η οποία στηρίζεται και παράγει αριθμούς σταθερής υποδιαστολής (fixed point) οι οποίοι μετατρέπονται σε κινητή (float). Η γεννήτρια ψευδοτυχαίων αριθμών δέχεται ένα αρχικό άνυσμα (seed) όπου με βάση αυτό το άνυσμα σε εσωτερικούς καταμετρητές (counters) αρχικά παράγονται κάποιοι αριθμοί οι οποίοι ανακατεύονται μέσω κυκλωμάτων ατάκτως με τέτοιο τρόπο ώστε να παράγουν αριθμούς σταθερής υποδιαστολής (fixed point). Αυτοί οι αριθμοί στέλνονται στην μονάδα (fixed to float unit) για να μετατραπούν σε κινητή υποδιαστολής (float) μορφή από όπου θα εισαχθούν στη μονάδα υπολογισμού του λογαρίθμου $\ln(x)$.

Η μονάδα $\ln(x)$ αρχικά είχε υλοποιηθεί με σειρά Taylor η οποία ναι μεν ήταν ακριβής αλλά καταλάμβανε τεράστιο χώρο στο FPGA και αύξανε τις εσωτερικές καθυστερήσεις (latencies). Για αυτό το λόγο σχεδιάστηκε μια καινούργια μονάδα η οποία υλοποιεί τον λογάριθμο με προσεγγίσεις πολυωνύμων 2^{ου} βαθμού (second order interpolation) το οποίο αποδείχθηκε οικονομικό ως προς το χώρο καθώς μειώνονται οι εσωτερικές καθυστερήσεις. Έπειτα όπως παρατηρούμαι και από την σχήμα (33) ο αριθμός που υπολογίστηκε προηγουμένως από τον λογάριθμο αφαιρείται για τον υπολογισμό του $-\ln(r_j)$ της εξίσωσης (25). Προχωρώντας στο κάτω δικτύωμα υπολογίζουμε το 1^ο κομμάτι της εξίσωσης (25) όπου αρχικά υπολογίζεται η πιθανότητα αντίδρασης $\alpha_j(x)$ ανάλογα με το βαθμό (reaction order) με βάση τον πίνακα 1 του Κεφαλαίου 2. Έπειτα υπολογίζεται ο πολλαπλασιασμός του πάνω δικτυώματος με το κάτω για τον τελικό υπολογισμό του χρόνου.

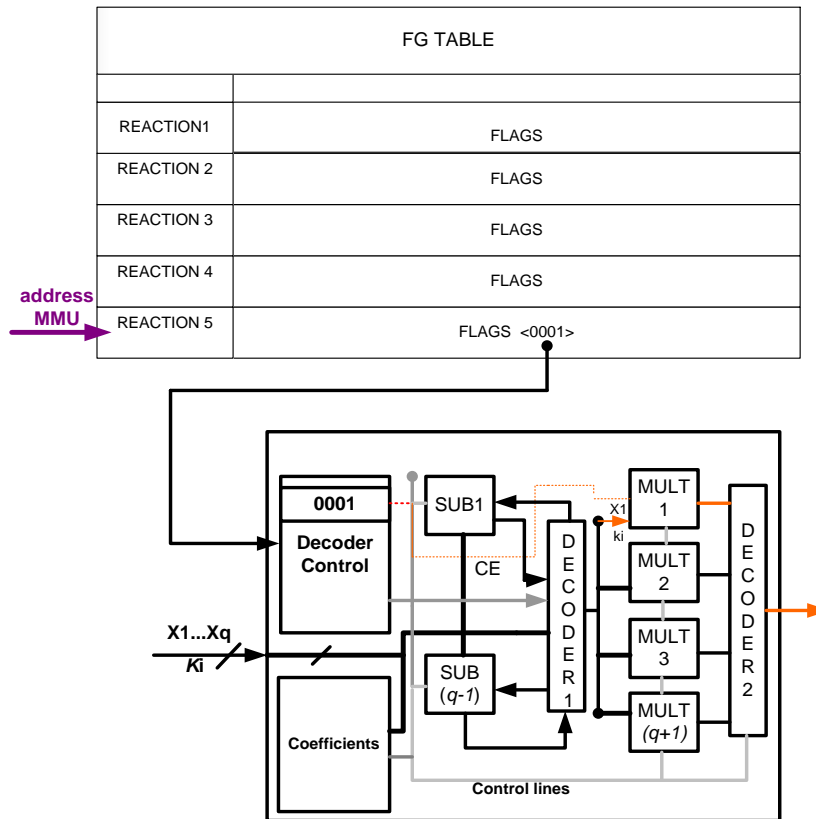


Σχήμα 33: Διάγραμμα απεικόνισης υπό-μονάδας PE για μέχρι 3^{ου} βαθμού αντιδράσεις.

Η μονάδα εκείνη που καθορίζει το κρίσιμο μονοπάτι είναι η μονάδα υπολογισμού της πιθανότητας αντίδρασης $\alpha_j(x)$ με βάση την ανάλυση του πίνακα 2 και αυτό γιατί επιδέχεται τις μεγαλύτερες καθυστερήσεις.

Η μονάδα αυτή δέχεται κάποια flags από το Flag Table τα οποία εισέρχονται σε έναν αποκωδικοποιητή, ο οποίος αναλόγως την περίπτωση του βιοχημικού μοντέλου ενεργοποιεί τις κατάλληλες υπό-μονάδες σε συνεργασία πάντα με την εσωτερική μονάδα ελέγχου της μονάδος PE όπως φαίνεται και στο διάγραμμα (σχήμα 34).

Η μονάδα αυτή μπορεί να παραμετροποιηθεί ως εξής: για q βαθμό (order) αντίδρασης θα υπάρχουν στο σύστημα $(q-1)$ αφαιρέτες και $(q+1)$ πολλαπλασιαστές, όπως επίσης θα πολλαπλασιάζεται πάντα στο τέλος του υπολογισμού του $\alpha_j(x)$ μία σταθερά από 1 μέχρι $1/2*q$. Βεβαίως δεν έχει βρεθεί από τη μέχρι τώρα ανάλυση των βιοχημικών αντιδράσεων αντιδράσεις με q μεγαλύτερο του 3 αλλά η διαδικασία αυτή φαίνεται να μπορεί να παραμετροποιείται και για μεγαλύτερους βαθμούς αντιδράσεων.



Σχήμα 34: Διάγραμμα απεικόνισης λειτουργίας και προγραμματισμού για τον υπολογισμό της πιθανότητας αντίδρασης.

Παρατηρώντας το σχήμα 34 βλέπουμε τη λειτουργία του FG πίνακα για την κάθε αντίδραση. Δίνουμε ένα παράδειγμα μιας συνηθισμένης 1^{ου} βαθμού αντίδρασης που είναι και η βέλτιστη περίπτωση καθώς παρατηρούνται οι μικρότεροι κύκλοι καθυστέρησης. Σε αυτό το παράδειγμα μπορούμε να προσομοιώσουμε μέχρι 3^{ου} βαθμού αντιδράσεις. Για αυτό το λόγο και το αποτέλεσμα της αλληλουχίας των bit είναι 4 δηλαδή (q+1) με q=3. Η MMU ανάλογα την αντίδραση την οποία τίθεται να επεξεργαστεί θα δώσει την ανάλογη διευθυνσιοδότηση σε όλους τους πίνακες (RT, VT, FG, DG). Ο αποκωδικοποιητής ελέγχου (Decoder control) δέχεται το άνωσμα των 4 bit από τον FG πίνακα. Ανάλογα την αλληλουχία που δέχεται εξαρτημένου των αντιδρώντων της αντίδρασης ενεργοποιεί αντίστοιχα τις κατάλληλες υπόμοναδες αφαιρετών και πολλαπλασιαστών (SUB και MULT). Ο στόχος είναι ανάλογα την περίπτωση να βρεθεί η πιθανότητα αντίδρασης με βάση τον πίνακα 1. Παράλληλα όλα τα δεδομένα εισόδου θα εισέλθουν στον Decoder 1. Τα σήματα ελέγχου θα δοθούν από την μονάδα Decoder control στον Decoder1 και αντίστοιχα στις μονάδες αριθμητικής δηλαδή για το παράδειγμα μας θα ενεργοποιηθεί μόνο ο Mult1 (κόκκινη γραμμή) για να γίνει η πράξη της 1^{ης} γραμμής του πίνακα 1. Ανάλογα και για τις άλλες περιπτώσεις του πίνακα 1 (κεφάλαιο 2) ενεργοποιούνται οι σωστές υπόμοναδες. Για την μονάδα παραγωγής ψευδοτυχαίων αριθμών έγινε ειδική σχεδίαση δημιουργίας κυκλώματος παραγωγής τυχαίων αριθμών σταθερής υποδιαστολής όπου με κατάλληλη

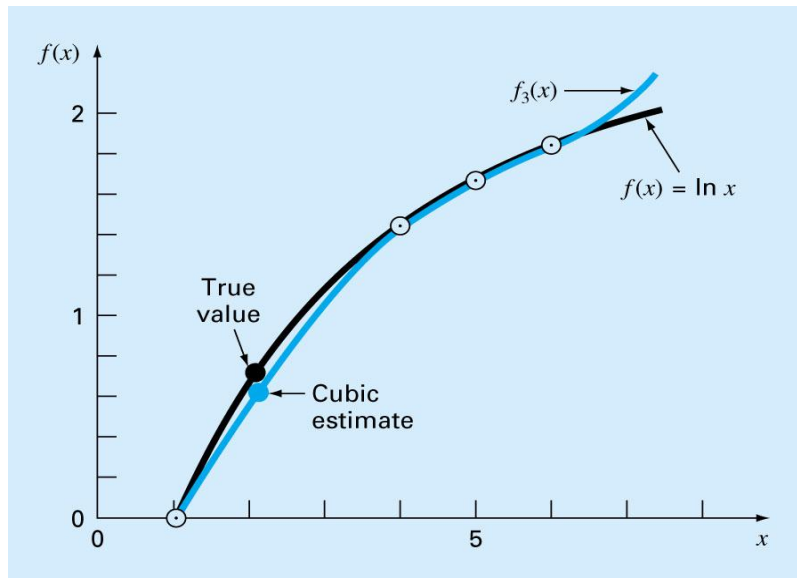
μετατροπή γίνεται διαμόρφωση σε αριθμούς κινητής υποδιαστολής από ειδική μονάδα. Πιο αναλυτικά η μονάδα (RNG) περιέχει απαριθμητές 16 bit οι οποίοι απαριθμούν με βάση ένα αρχικό άνυσμα (seed) το οποίο εισέρχεται μέσω του προγραμματισμού του συστήματος από τον χρήστη. Κατόπιν γίνονται ειδικής μορφής αντιμεταθέσεις των bits για την παραγωγή τυχαίων αριθμών σταθερής υποδιαστολής από (0 μέχρι 1). Η σχεδίαση της μονάδας αυτής έχει περάσει τα Tests τυχαιότητας (Die Hard) συγκεντρώνοντας τα διανύσματα σε ένα πίνακα δεδομένων και ελέγχοντας την τυχαιότητα μέσω κώδικα σε C.

Επίσης για τις πολλαπλές προσομοιώσεις υπάρχει μηχανισμός ο οποίος κρατάει την τελευταία απαρίθμηση που έγινε μέχρι το τέλος της πρώτης προσομοίωσης, ο οποίος αριθμός περνάει σαν αρχικό άνυσμα (seed) στους απαριθμητές ώστε στην έναρξη της δεύτερης προσομοίωσης να αρχίσει η απορύθμιση με την συγκεκριμένη αρχική τιμή.

Για την παραγωγή του λογαρίθμου έγιναν ειδικές μελέτες και σχεδιάσεις σε Xilinx. Σε σχεδιάσεις σε Altera χρησιμοποιήθηκε ειδική μονάδα παραγωγής του λογαρίθμου κινητής υποδιαστολής από την βιβλιοθήκη (Mega Wizard του Quartus). Στην σχεδίαση σε Xilinx δημιουργήσαμε μια μονάδα ειδική για την παραγωγή του λογαρίθμου αρχικά βασιζόμενη σε σειρά Taylor της μορφής.

$$\ln(x) = (x - 1) + \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} + \dots \sum_{n=1}^{\infty} \frac{(-1)^n (x-1)^n}{n} \quad (26)$$

Όπου όσο το n αυξάνει ως προς το βαθμό τόσο αυξάνει και ο βαθμός ακρίβειας του λογαρίθμου. Αυτή η σχεδίαση όμως δεν αποδείχθηκε οικονομική από μεριάς χωρητικότητας και συχνότητας στο FPGA. Πιο συγκεκριμένα επειδή χρησιμοποιήθηκαν blocks κινητής υποδιαστολής παρατηρήθηκε ότι όσο μεγαλώνει η ακρίβεια τόσο αυξάνει και η κατάληψη επιφάνειας (hardware) στο FPGA. Για αυτό το λόγο έγινε μελέτη και σχεδίαση κυκλώματος προσέγγισης του λογαρίθμου με βάση τα πολυώνυμα δευτέρου βαθμού (second order interpolation) όπως φαίνεται και από το σχήμα (35).



Σχήμα 35: Προσέγγιση λογαρίθμου με βάση πολυωνμική εξίσωση 2^{ου} βαθμού.

Η περιγραφή της γενικής εξίσωσης είναι η ακόλουθη :

$$P(x) = a_0 + a_1(x - c)^2 + a_2(x - c)^2 + \dots + a_n(x - c)^n \quad (27)$$

Η Ανάλυση της πολυωνμικής εξίσωσης κατά Newton μας δίνει την παρακάτω γενική εξίσωση

$$P(x) = a_0 + a_1(x - c_1) + a_1(x - c_1)(x - c_2) + \dots + a_n(x - c_1)(x - c_2) \dots (x - c_n) \quad (28)$$

Όπου η παραμετροποίηση του βαθμού ακρίβειας έγκειται στο ότι για την περιγραφή μιας οποιαδήποτε πολυωνμικής ανάλυσης κατά Newton χρειάζονται $n + \frac{n(n+1)}{2}$ μονάδες άθροισης και $\frac{n(n+1)}{2}$ μονάδες πολλαπλασιασμού. Με βάση τα πολυώνυμα 2^{ου} βαθμού μειώθηκαν οι απαιτήσεις σε επιφάνεια στο FPGA σε σύγκριση με την σειρά Taylor και μειώθηκαν επίσης και οι εσωτερικές καθυστερήσεις στο διπλάσιο.

3.8 Στατιστικά στοιχεία προσομοιώσεων σε Xilinx και Altera για τα SoCs FRM1X, FRM2X, FRM4X, FRM8X

Στο κομμάτι αυτό θα παρουσιάσουμε τα στατιστικά στοιχεία μετά από Post Place and Route προσομοίωση δηλαδή στατιστικά στοιχεία από τον XST και το Quatrus στα οποία έχει γίνει πλήρες σύνθεση του κυκλώματος για το συγκεκριμένο διαθέσιμο FPGA.

Αρχίζοντας την ανάλυση από τον Πίνακα 2 παρατηρούμαι τα στατιστικά στοιχεία για την βασική υπό-μονάδα PE των SoCs η οποία δέχεται και όλο το βάρος της επεξεργασίας και καθορίζει την συχνότητα λειτουργίας. Παρατηρούμαι για τη κάθε υπό-μονάδα PE τους κύκλους καθυστέρησης, την κατανομή σε logic elements (LUTS), καταχωρητές (Regs), μονάδες DSP και την δέσμευση σε Block RAMs.

Για την μονάδα αυτή τόσο για την υλοποίηση της σε Altera αλλά και σε Xilinx

χρησιμοποιήθηκαν έτοιμα κομμάτια από τις βιβλιοθήκες των εργαλείων και πιο συγκεκριμένα από βιβλιοθήκες που περιέχουν μονάδες αριθμητικής για πράξεις κινητής υποδιαστολής (floating point blocks IEEE 754 single precision).

Πίνακας 2: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα PE

Xilinx (Virtex 5 XC5VLX110T-1ff1136 – Virtex 7 XC7V855T-3ffg1157)					
Component	Frequency (MHz)	Latency (cycles)	LUTS	DSPs	Regs
RNG	448*- 665**	2	28	0	8
FIX TO FLOAT	368*- 683**	3	211	0	254
LN(x)	301*- 489**	22	1573	0	1306
SUB	300*- 683**	8	267	2	379
MULT	345*- 642**	8	655	0	707
CU	472 * - 839**	1	87	0	120
Aj(x)	300*- 392**	Q [8-48]	3965	2	4122
DIVISION	371*- 392**	14	783	0	1388
PE	297*- 344**	44-70	7669	4	8290

* Μέγιστη συχνότητα λειτουργία για το Virtex 5

** Μέγιστη συχνότητα λειτουργία για το Virtex 7

Επίσης κατά τη διάρκεια της σχεδίασης και με κύριο γνώμονα την αύξηση της συχνότητας λειτουργίας και την ελάττωση των εσωτερικών καθυστερήσεων για την κάθε μονάδα αριθμητικής λογικής, που πάρθηκε από τις βιβλιοθήκες IEEE 754 έγινε ειδική μελέτη για την επιλογή του κατάλληλου τρόπου σωλήνωσης (pipelining). Στις μονάδες αυτές τα εργαλεία σύνθεσης επιτρέπουν στον χρήστη να διαμορφώσει την υπό μονάδα ανάλογα με τις απαιτήσεις της σχεδίασης σε συχνότητα και με βάση την κατανομή χώρου που διατίθεται στο FPGA. Όπως παρατηρούμαι έγινε και μελέτη για την αύξηση της απόδοσης για τα διάφορα FPGAs νεώτερης τεχνολογίας για όλες τις σχεδιάσεις. Ο πίνακας 2 παραθέτει πληροφορίες για την μονάδα της γεννήτριας ψευδοτυχαίων αριθμών (Random Number Generator), την μονάδα μετατροπής από αριθμητική σταθερής υποδιαστολής (fixed point) σε float, την μονάδα λογαρίθμου όπου για Xilinx έγινε ειδική σχεδίαση με προσεγγίσεις πολυωνύμων 2^{ου} βαθμού (second order interpolation) ενώ για την σχεδίαση σε Altera η μονάδα χρησιμοποιήθηκε απευθείας από την βιβλιοθήκη. Επίσης παραθέτονται τα στατιστικά στοιχεία για τη μονάδα αφαίρεσης SUB, πολλαπλασιασμού (MULT), της μονάδας ελέγχου CU για τον επεξεργαστή PE και για την μονάδα υπολογισμού της πιθανότητας αντίδρασης $a_j(x)$. Η μεταβλητή Q στους κύκλους καθυστέρησης υποδηλώνει ότι οι καθυστερήσεις εξαρτώνται από τον βαθμό της αντίδρασης (reaction order) και το είδος της αντίδρασης. Έτσι για εξισώσεις 3^{ου} βαθμού και με βάση τον Πίνακα 1 του κεφαλαίου 2 οι μεγαλύτερες καθυστερήσεις παρατηρούνται όταν και τα 3 στοιχεία της

αντίδρασης είναι ίδια. Η τελευταία σειρά του Πίνακα 2 δίνει την συνολική πληροφορία για όλη την μονάδα. Ο πίνακας 3 παραθέτει τα στατιστικά στοιχεία για την μονάδα FRM1X με ένα PE για σχεδίαση σε Xilinx για τα FPGA της κατηγορίας Virtex.

Πίνακας 3: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα FRM1X

Xilinx (Virtex 5 XC5VLX110T-1ff1136 – Virtex 7 XC7V855T-3ffg1157)						
Component	Frequency (MHz)	Latency (cycles)	LUTS	DSPs	Regs	BRAMS
CU	301*-406*	2	1811	0	1988	0
MMU	952.381	4	1504	0	512	0
RT	379	1	0	0	0	8
VT	379	1	0	0	0	8
ST	379	1	4	0	0	25
FT	379	1	0	0	0	2
PE	297*-344**	Q [44-70]	7669	4	7930	1
MTU1	335*-564 **	1	73	0	41	0
TU	302*-385**	8	504	2	516	1
SU	300*-683**	8	1068	8	1516	4
FRM1X	297*-320**	[44-70]+9	24779	12	12942	49

* Μέγιστη συχνότητα λειτουργία για το Virtex 5

** Μέγιστη συχνότητα λειτουργία για το Virtex 7

Από τον πίνακα 3 παρατηρούμαι ότι το κρίσιμο μονοπάτι είναι η μονάδα PE η οποία καθορίζει την απόδοση της σχεδίασης. Με την σειρά παραθέτονται πληροφορίες για όλες τις βασικές υπό μονάδες της FRM σχεδίασης, όπως για την Control Unit, την MMU (Memory Management Unit), τις μνήμες RT,VT,ST,FG(Flag Table) καθώς και τις μονάδες PE, TU (Time Update) και SU (Species Update). Η τελευταία γραμμή παραθέτει στο σύνολο τα στατιστικά στοιχεία για την FRM1X σχεδίαση.

Προτού αναλύσουμε τις σχεδιάσεις με περισσότερα PEs θα δείξουμε με βάση την ανάλυση σε τομείς που έγινε στο κεφάλαιο 3 τα στατιστικά στοιχεία για τους κύριους αυτούς 3 τομείς οι οποίοι υπάρχουν σε όλες τις σχεδιάσεις για υλοποιήσεις σε FPGA της κατηγορίας Virtex.

Πίνακας 4: Στατιστικά στοιχεία για τις βασικές υπό-μονάδες των FRM4X_NRM4X SoC

Xilinx (Virtex 5 XC5VLX110T-1ff1136 – Virtex 7 XC7V855T-3ffg1157)						
Core	Frequency (MHz)	Latency (cycles)	LUTS	DSPs	Regs	RAMS
MMU/CU	400	7	2731	4	2442	12
DSP	338	44-70	15748	48	52032	2
Mem Subsystem	320	13	8809	0	5291	89

Από τον Πίνακα 4 παρατηρούμαι ότι τις μεγαλύτερες απαιτήσεις σε χώρο της έχει η μονάδα DSP λόγω των μονάδων κινητής υποδιαστολής οι οποίες καταλαμβάνουν πολύ χώρο στο FPGA. Όπως επίσης παρατηρείται και αύξηση στους κύκλους καθυστέρησης της μονάδας αυτής.

Ο πίνακας 5 ο οποίος είναι και μείζονος σημασίας δείχνει τα στατιστικά στοιχεία

προσομοιώσεων για όλες τις σχεδιάσεις όπου αναφέρονται οι επιδόσεις του συστήματος για 1,2,4 μέχρι και 8 PEs. Παρατηρούμαι ότι για ένα σταθερό μοντέλο 512 αντιδράσεων η απόδοση σε Mreactions/sec αυξάνεται σε τρόπο λειτουργίας '1' καθώς αυξάνεται η παραλληλοποίηση των συστημάτων, πράγμα που καθιστά τον FRM αλγόριθμο τον πλέον κατάλληλο για προσομοίωση μεγάλων βιοχημικών μοντέλων.

Πίνακας 5: Στατιστικά στοιχεία P.P.R προσομοίωσης και αναφορά επιδόσεων, εσωτερικών καθυστερήσεων και κατανομής χώρου για τα FRM1X-2X -4X-8XSoC.

Xilinx (Virtex 5 XC5VLX110T-1ff1136 – Virtex 7 XC7V855T-3ffg1157)				
	FRM1X	FRM2X	FRM4X	FRM8X
Reactions	512	1024	2048	4096
Frequency (MHz)	200-320	200-320	200-320	200-320
Latency (cycles)	Q (44-70)+9	Q (44-70)+11	Q (44-70)+13	Q (44-70)+15
Performance (Mreactions/sec)	173- 181*	303-526*	485-776 *	687-1099*
	292- 302**	329-548**	553-922**	832-1332**
LUTs	24779	36919	54557	71641
DSPs	12	24	58	64
REGS	15720	30650	57323	72258
RAM-BLOCKS	53	118	122	141

Μέγιστη συχνότητα λειτουργίας για το Virtex 5 στα 200 MHz

Μέγιστη συχνότητα λειτουργίας για το Virtex 7 στα 320 MHz.

* Ανάλυση για 3 αντιδρώντα του ίδιου στοιχείου (σημειώνονται οι μεγαλύτερες καθυστερήσεις)

**Ανάλυση για ένα ασταθές αντιδρών (1 στοιχείο το οποίο διασπάται στα υπό προϊόντα του)

Οι παρακάτω πίνακες δείχνουν τις αντίστοιχες υλοποιήσεις για τα αναπτυξιακά της Altera, όπου η σχεδίαση έγινε στο προσομοιωτή Quartus και χρησιμοποιήθηκε το FPGA Cyclone III. Όπως θα δούμε και στους πίνακες οι υλοποιήσεις σε Altera αν και προσφέρουν στον χρήστη μια μεγάλη ευκολία στη δημιουργία σχεδιάσεων με μονάδες κινητής υποδιαστολής παρατηρείται μείωση στη συχνότητα λειτουργίας το οποίο αποδείχθηκε μεγάλο μειονέκτημα για την δημιουργία αποδοτικότερων κυκλωμάτων. Αυτό συμβαίνει γιατί η βιβλιοθήκη της Altera (Mega Wizard Library) αν και διαθέτει μια πληθώρα από έτοιμα components δεν αφήνει στον χρήστη την δυνατότητα παρεμβολής και ρύθμισης της σωλήνωσης (pipelining) των μονάδων αυτών όπως στη Xilinx. Αυτός ήταν και ο κύριος λόγος που οι επόμενες βελτιωμένες σχεδιάσεις βασίστηκαν επί το πλείστον σε σχεδιάσεις που έγιναν για FPGA της Xilinx.

Ο πίνακας 6 αναλογικά με τον πίνακα 2 παραθέτει τα αντίστοιχα στοιχεία για την σχεδίαση της μονάδος PE. Παρατηρούμαι επίσης τις διαφορές στην κατανομή των διαφόρων βασικών μονάδων στα FPGA. Τα LUTs της Xilinx με τα LEs της Altera είναι η ίδια βασική μονάδα αντιστοίχισης όπως φαίνεται και από το σχήμα 36 και αποτελείται από το Look Up Table που περιγράφει την λογική σχεδίαση τον πολυπλέκτη επιλογής και τους καταχωρητές. Αντιστοιχώς το ίδιο ισχύει και για τα Logic Elements της Altera. Διαφορές όμως παρουσιάζονται στον τρόπο ανάλυσης των Block Rams. Πιο συγκεκριμένα η Xilinx χρησιμοποιεί την κατανομή σε Block Rams και αντιστοιχεί τις μνήμες με βάση των κώδικα

που γράφεται ενώ η Altera σε memory bits που αντιστοιχούν σε συγκεκριμένες θέσεις αποθήκευσης σαν Block Rams στο FPGA. Επίσης μια άλλη διαφορά είναι ότι η σύνθεση σε Altera μετράει την συνδυαστική λογική κυρίως για τις FSM υλοποιήσεις και παραθέτει για την συνδυαστική αυτή λογική ειδικά blocks που τα ονομάζει Combinational function blocks. Αντίστοιχα η Xilinx είτε χρησιμοποιεί Block Rams ή (slice registers) και τα δεσμεύει ως καταχωρητές. Επίσης μια άλλη εμφανής διαφορά είναι η υλοποίηση των αντίστοιχων μονάδων μαθηματικών πράξεων σε DSP blocks για την Xilinx και αντίστοιχα σε embedded Booth type πολλαπλασιαστές για την Altera.



Σχήμα 36: Μονάδα LUT Xilinx.

Πίνακας 6: Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα PE σχεδίαση σε Altera για 2^{ου} βαθμού αντιδράσεις.

ALTERA CYCLONE III EP3C120F780C8							
Component	Frequency (MHz)	Latency (Cycles)	LEs	Embedded Mult	Regs	Memory (bits)	Comb functions
RNG	129	14	775	0	335	0	722
LN(x)	147.36	21	2793	6	1755	493	2298
SUB	133.9	7	802	0	343	36	782
Mult	158.4	5	320	7	217	0	242
AJ(x)	158.4	10	640	14	434	0	484
Division	112	33	6914	0	3126	845	4784
τ -processor (PE)	112	47	12444	27	6410	1374	9612

Ο πίνακας 7 εν αντιστοιχία με τον Πίνακα 3 παραθέτει για την FRM1A σχεδίαση (FRM1X για Xilinx FRM1A για Altera) την αντιστοιχία σε στατιστικά στοιχεία μετά και από P.P.R προσομοίωση για την κάθε βασική μονάδα και για το σύνολο με 1 PE. Αυτό που παρατηρείται είναι η μείωση στη συχνότητα λειτουργίας και άρα της απόδοσης. Η σχεδίαση σε Altera έγινε κυρίως για ερευνητικούς σκοπούς και πιο συγκεκριμένα για να διαπιστωθεί εάν θα μπορούσε να υπάρξει μεγαλύτερη απόδοση με άλλου τύπου FPGA, για αυτό και για τις ίδιες σχεδιάσεις χρησιμοποιήθηκε το απλό μοντέλο των 512 αντιδράσεων 2^{ου} βαθμού.

Πίνακας 7 : Στατιστικά στοιχεία μετά από P.P.R για την υπό-μονάδα FRM1A σχεδίαση σε Altera.

ALTERA CYCLONE III EP3C120F780C8							
Component	Frequency (MHz)	Latency (Cycles)	LEs	Embedded Multipliers	Regs	Memory (bits)	Comb functions
CU	160	2	394	0	347	0	231
MMU	225	1	119	0	109	0	56
RT	170	1	0	0	0	81920	0
VT	170	1	0	0	0	65536	0
ST	170	1	0	0	0	524288	0
τ -processor (PE)	112	47	12444	27	6410	1374	9612
MTU	164	1	88	0	41	0	66
TU	133.9	7	874	0	461	36	790
SU	133.9	7	802	0	343	144	782
FRM1A	112	54	16020	27	9415	692609	13042

Ο πίνακας 8 δίνει για τις αντίστοιχες σχεδιάσεις με πολλά PEs τις αποδόσεις των συστημάτων, όπου και πάλι παρατηρείται για αύξηση των PEs και για παραλληλισμό του FRM αλγορίθμου, αύξηση της απόδοσης για ένα σταθερό μοντέλο 512 αντιδράσεων.

Πίνακας 8: Στατιστικά στοιχεία P.P.R προσομοίωσης για επιδόσεις Mreactions/sec, εσωτερικές καθυστερήσεις και απόδοσης (MRC/sec) για τα FRM1X-1A -2A-4A SoCs σχεδίαση σε Altera

ALTERA CYCLONE III EP3C120F780C8			
	FRM1A	FRM2A	FRM4A
Reactions	512	1024	2048
Frequency (MHz)	112	112	112
Latency (cycles)	576	578	580
Performance (MReactions/sec)	128	256	512
LEs	16020 (13%)	32645 (27%)	64644 (54%)
Embedded Multipliers	27 (5%)	54 (9%)	108 (18.7%)
Registers	9415	19211	38068
Memory (bits)	692609 (17%)	1277952 (32%)	2490368 (62.5%)
Comb functions	13042 (11%)	26585 (22%)	52743 (44.28%)

ΚΕΦΑΛΑΙΟ 4

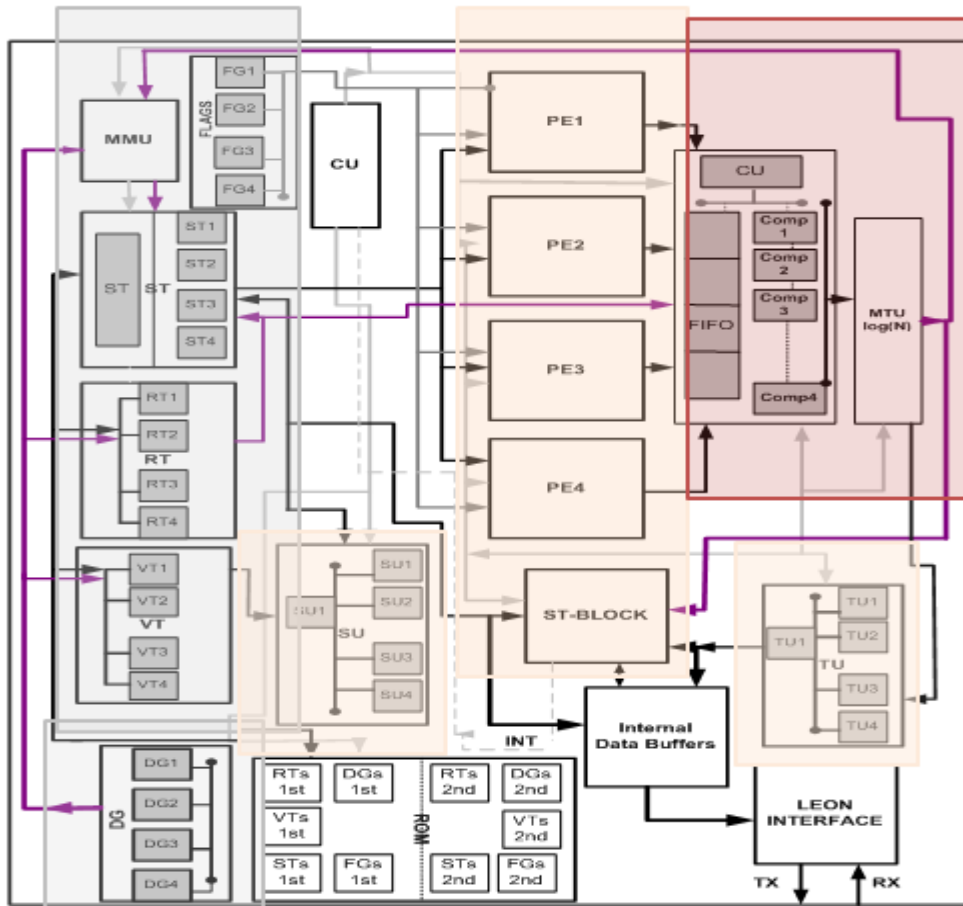
Αρχιτεκτονική FRM-NRM SoC

4.1 Γενική περιγραφή του FRM4X-NRM4X SoC

Στην παράγραφο αυτή θα αναλυθεί η σχεδίαση για την αρχιτεκτονική που επιτρέπει την υλοποίηση δύο αλγορίθμων σε ένα SoC του FRM και NRM. Όπως μπορούμε να παρατηρήσουμε από το σχήμα 37 και με βάση και τις προηγούμενες αναλύσεις χωρίζουμε την αρχιτεκτονική στους κύριους τομείς. Με γκρι πλαίσιο πάλι φαίνεται ο τομέας διαχείρισης μνήμης (memory subsystem), ο τομέας αριθμητικών υπολογιστών (DSP system) με πορτοκαλί πλαίσιο και οι τομείς εύρεσης μικρότερου χρόνου Minimum Time Unit με κόκκινο. Ο κάθε τομέας απαρτίζεται από σχεδόν τις ίδιες υπό-μονάδες σε σχέση με την υλοποίηση που αναλύσαμε όταν υπήρχε μόνο η αρχιτεκτονική FRM4X. Αυτός ήταν ο λόγος της δυνατότητας ύπαρξης σε ένα chip δύο αλγορίθμων. Υπάρχουν βέβαια αλλαγές στις λειτουργίες κάποιων βασικών βαθμίδων σε σχέση με τις FRM υλοποιήσεις.

Όπως παρατηρούμαι έχουν ενταχθεί νέες μονάδες και έχουν προστεθεί παραπάνω λειτουργίες στην μονάδα ελέγχου καθώς και στον τρόπο διευθυνσιοδότησης της MMU.

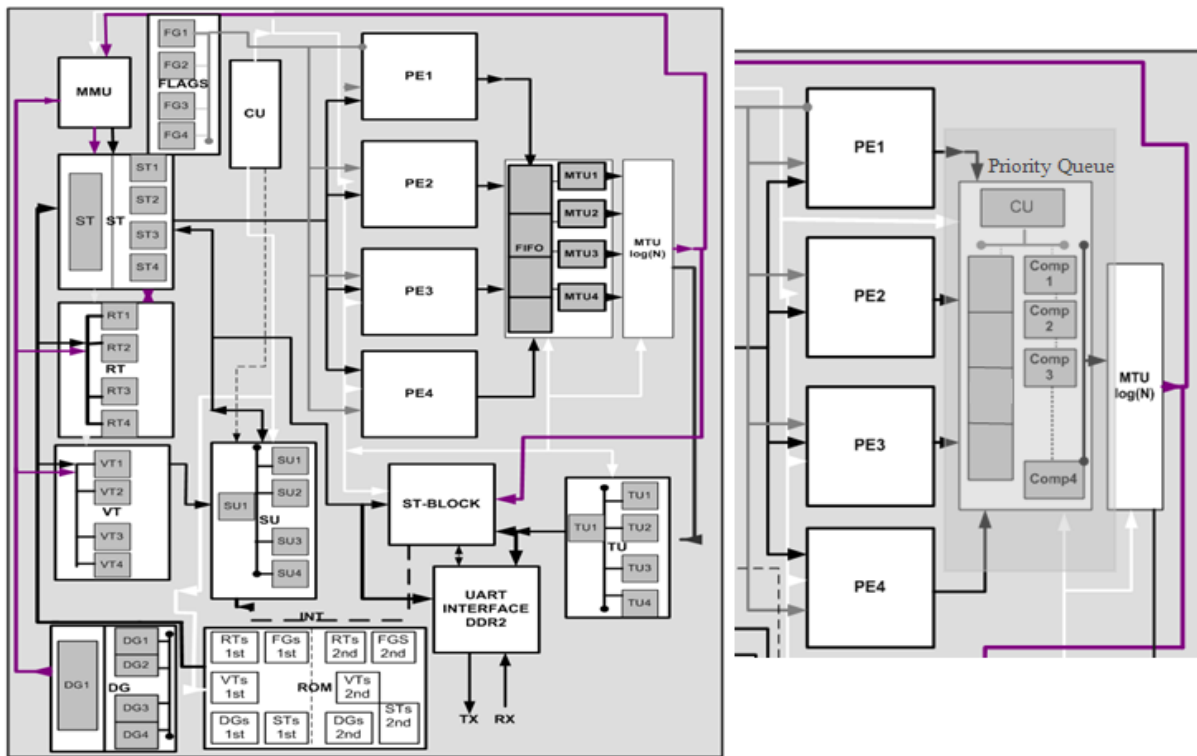
Πιο αναλυτικά στο γκρι πλαίσιο που αφορά την οργάνωση των μνημών (memory sub – system) έχει προστεθεί ο γράφος εξαρτήσεων (dependency graph) ο οποίος περιέχει τις σχέσεις εάν υπάρχουν μεταξύ των αντιδράσεων, δηλαδή το ποιά στοιχεία της μιας αντίδρασης επηρεάζουν την άλλη. Επίσης και στο τμήμα MTU υπάρχουν μνήμες προσωρινής κράτησης δεδομένων των χρόνων $\tau_j(x)$ και των διευθύνσεων καθώς και δομές κράτησης δεδομένων με σειρά προτεραιότητας από το μεγαλύτερο στο μικρότερο Priority Queues όπου αποθηκεύονται οι χρόνοι $\tau_j(x)$ και οι αντίστοιχες διευθύνσεις.



Σχήμα 37: Διάγραμμα αρχιτεκτονικής FRM-NRM με 4 επεξεργαστές.

Έχουν υλοποιηθεί για τον NRM αλγόριθμο 3 διαφορετικές σχεδιάσεις η 1^η χρησιμοποιεί μία δομή FIFO για την αποθήκευση των δεδομένων $\tau_j(x)$ και των διευθύνσεων σε συγκεκριμένες θέσεις (σχήμα 38). Αυτοί οι χρόνοι θα δοθούν από το σημείο αυτό αρχικά στην μονάδα των MTUs και έπειτα στην δένδροειδής δομή του MTU2. Η 2^η σχεδίαση χρησιμοποιεί έναν μηχανισμό παράλληλης ταξινόμησης των χρόνων από το μικρότερο στο μεγαλύτερο ώστε στα N τμήματα των πινάκων να βρίσκετε η μικρότερη διεύθυνση. Ο 3^{ος} τρόπος χρησιμοποιεί δένδρα προτεραιότητας τα οποία και συνθέτει σε ένα πίνακα (κόκκινο πλαίσιο σχήμα 37). Τα δεδομένα που έρχονται από τα PEs αποθηκεύονται σε συγκεκριμένα μέρη σε ένα πίνακα όπου έπειτα διαμορφώνονται τα δέντρα (Heap trees). Οι μέθοδοι κατηγοριοποιήσεις και εύρεσης του μικρότερου χρόνου θα αναλυθούν σε ειδική ενότητα στο παρών κεφάλαιο. Επίσης παρατηρούμαι ότι για να υποστηριχθούν και οι 2 τρόποι λειτουργίας και για τους 2 αλγορίθμους οι μνήμες DG, ST και PQ χωρίζονται σε 2 τομείς όπου ο ένας τρόπος χρησιμοποιεί 1 πίνακα (array) και ο άλλος N τομείς πινάκων. Πάλι μέσω του SysPy και ανάλογα με το μοντέλο αρχικοποιούνται όλες οι μνήμες καθώς και οι κρυφές μνήμες που κρατούν τις αρχικές συγκεντρώσεις για τη μνήμη ST για την δυνατότητα ύπαρξης πολλαπλών προσομοιώσεων.

4.2 Περιγραφή όλων των υπομονάδων FRM4X-NRM4X SoC



Σχήμα 38: Περιγραφή βασικών υπό μονάδων FRM-NRM με 4 επεξεργαστές.

Η ανάλυση μας ξεκινά με την διαδρομή των δεδομένων για τον NRM. Η πιο δύσκολη λειτουργία στην περίπτωση του NRM ήταν η παραλληλοποίηση και η παραμετροποίηση του γράφου εξαρτήσεων. Το μυστικό ήταν το κατάλληλο χώρισμα του γράφου και το σύστημα διευθυνσιόδοτησης του. Ο γράφος αυτός χωρίζεται σε N μέρη, η κάθε σειρά αντιστοιχεί σε κάθε αντίδραση. Η κάθε σειρά απαρτίζεται από 10 οχτάδες που περιέχουν 8 αριθμούς σε hex στην μορφή που είδαμε στις RT μνήμες όπου όλοι οι 8 αριθμοί είναι διευθύνσεις που αντιστοιχούν σε εξαρτήσεις δηλαδή στις αντιδράσεις που επηρεάζονται από την πυροδότηση της μίας. Άρα η κάθε γραμμή του πίνακα του κάθε εκ των N τομέων, αντιστοιχεί στις εξαρτήσεις που μπορεί να έχει η κάθε αντίδραση και είναι 10×8 , δηλαδή μέχρι 80 στοιχεία μπορούν να επηρεάζονται από την κάθε αντίδραση για την συγκεκριμένη αρχιτεκτονική. Οι διευθύνσεις λοιπόν σε αυτόν τον πίνακα χωρίζονται ανά 10 και έτσι κάθε γραμμή του πίνακα αντιστοιχεί σε μια αντίδραση. Όποτε η MMU στέλνει μία διεύθυνση, όπου αυτή η διεύθυνση είναι χαρακτηριστική της κάθε αντίδρασης και συνεπώς η επόμενη διεύθυνση θα αθροίζεται κατά ένα παράγοντα μετατόπισης 10 για να αντανακλά η κάθε αντίδραση σε κάθε γραμμή του πίνακα. Η μονάδα ελέγχου CU προγραμματίζεται από τον χρήστη σε τρόπο λειτουργίας NRM. Έστω ότι προγραμματίστηκε η μονάδα στον 1^ο τρόπο λειτουργίας δηλαδή μία N παράλληλη προσομοίωση. Η διαδικασία προσομοίωσης αρχίζει με τον ίδιο τρόπο όπως στον FRM. Δηλαδή η MMU στέλνει τις διευθύνσεις στις N RT

μνήμες από όπου αυτές αντιστοίχως στέλνουν διευθύνσεις που αντιστοιχούν στα στοιχεία της αντίδρασης με τις αρχικές συγκεντρώσεις. Οι ST μνήμες θα στείλουν τα δεδομένα στα PEs όπως επίσης και η Flag Table μνήμη θα στείλει τα δεδομένα στα PEs για να υπολογισθεί η πιθανότητα αντίδρασης. Τα PEs με βάση τις συγκεντρώσεις και τα (reaction rates, k) και έχοντας τα κατάλληλα flags θα υπολογίσουν τους χρόνους αντίδρασης. Από τα PEs για την κάθε αντίδραση θα φεύγουν προς τις μνήμες PQs (Priority Queues) τα $\tau_j(x)$ και $a_j(x)$.

Αναλυτικότερα το κάθε PE δίνει τα δεδομένα του με τη σειρά των αντιδράσεων που το αντιστοιχούν αρχικά σε μία N χωρισμένη FIFO. Από εκεί για το κάθε τμήμα N της FIFO δίδονται τα δεδομένα στα MTUs. Έτσι το κάθε κομμάτι της FIFO δίνει στα αντίστοιχα κομμάτια MTU1 τα δεδομένα όπου υπολογίζεται ο ελάχιστος χρόνος. Από όλα τα MTU1 οδηγούμαστε στο MTU2 για να βρεθεί στην περίπτωση της μιας προσομοίωσης ο ελάχιστος χρόνος και η αντίστοιχη διεύθυνση. Πάλι από έναν μηχανισμό όπως και στον FRM υπολογίζεται με βάση ένα σύστημα αποπολυπλεκτών και καταχωρητών συνεργαζόμενο με τις μονάδες MTU2 από ποιο RT εκ των N έχει προκύψει η ελάχιστη διεύθυνση από όπου από αυτό το κομμάτι θα δοθούν οι διευθύνσεις στην MMU, που αντιστοιχούν σε θέσεις στην ST μνήμη από όπου θα υπολογισθούν οι νέες συγκεντρώσεις από τους $(q+p)$ αθροιστές της μονάδος SU, ενώ ταυτόχρονα θα γίνει και η ανανέωση του χρόνου από μία εκ των N TUs (Time Update Units).

Η αντίδραση που κέρδισε και πιο συγκεκριμένα η διεύθυνση που αντιστοιχεί στην αντίδραση που κέρδισε θα αντιστοιχηθεί στην διεύθυνση του DG που βρίσκεται η πληροφορία που φέρει τις εξαρτήσεις της αντίδρασης που κέρδισε. Έτσι στον επόμενο κύκλο δεν θα γίνουν ξανά όλοι οι υπολογισμοί των $\tau_j(x)$ και $a_j(x)$ τα οποία βρίσκονται ήδη αποθηκευμένα στα PQs αλλά η MMU θα πάρει τις διευθύνσεις που αντιστοιχούν στις αντιδράσεις εκείνες που επηρεάστηκαν. Αυτές τις αντιδράσεις η MMU θα της μοιράσει διά N και θα ενημερώσει κατάλληλα τα N RTs με βάση τις διευθύνσεις που πήρε από τα DG. Από εκεί τα RTs θα δώσουν στην ST τις διευθύνσεις από όπου θα περάσουν τα δεδομένα στα PEs και θα υπολογισθούν κατάλληλα τα $\tau_j(x)$ και $a_j(x)$. Τα δεδομένα των $\tau_j(x)$ και $a_j(x)$ με βάση την αρχική διεύθυνση που αντιστοιχεί στον αντίστοιχο τομέα εκ των N από όπου είχε βρεθεί ο ελάχιστος χρόνος και με βάση τις διευθύνσεις του DG για τον τομέα αυτό, θα γίνει εγγραφή των $\tau_j(x)$ στον αντίστοιχο τομέα της FIFO. Από εκεί ξανά για όλα τα στοιχεία των PQs θα βρεθεί ο ελάχιστος χρόνος και η διεύθυνση που αντιστοιχεί σε αυτόν και έτσι η διαδικασία θα επαναληφθεί.

Όπως καταλαβαίνουμε με αυτόν τον τρόπο γλυτώνουμε ένα πολύ σημαντικό βήμα τον

υπολογισμό όλων των $\tau_j(x)$ και $a_j(x)$ από την αρχή, το οποίο έχει σαν αποτέλεσμα την αύξηση της απόδοσης σε σχέση με τον FRM όπου ενισχύεται ακόμα πιο πολύ από το γεγονός ότι παραλληλοποιείται διά N η διαδικασία. Δηλαδή εάν από μία αντίδραση η οποία πραγματοποιείται επηρεάζονται από ένα μεγάλο μοντέλο των 512 αντιδράσεων οι 80, θα υπολογισθούν μόνο τα $\tau_j(x)$ και $a_j(x)$ για αυτές τις 80 οι οποίες θα παραλληλοποιηθούν και θα διαιρεθούνε διά 4 στα PEs οπότε ο φόρτος εργασίας μειώνεται στις 20 αντιδράσεις.

Στην περίπτωση που χρησιμοποιούμαι μνήμη PQ αντί για FIFO χρειαζόμαστε έναν μηχανισμό γρήγορης ταξινόμησης. Με το PQ δεν χρειάζονται να συγκριθούν σειριακά όλοι οι M/N χρόνοι απλά με έναν γρήγορο τρόπο ταξινόμησης που θα αναλυθεί σε επόμενη ενότητα στην κορυφή των N τμημάτων του PQ βρίσκονται πάντοτε οι μικρότεροι χρόνοι άρα στα επόμενα στάδια της δενδροειδής δομής συγκριτών θα έχω από του 4 μικρότερους χρόνους την εύρεση του μικρότερου.

Στην περίπτωση παράλληλων προσομοιώσεων η διαδικασία χωρίζεται σαν να υπάρχουν στο σύστημα $N \times NRM1X$ παράλληλα συστήματα. Η αρχικοποίηση θα γίνει από το SysPy για N διαφορετικά μοντέλα για τις μνήμες ST, RT, VT, Flag Table και DG. Σε αυτό το σημείο αξίζει να σημειωθεί ότι η SBML δεν δίνει πληροφορίες για τις εξαρτήσεις σε μορφή κάποιου txt αρχείου, απλά αυτές υπολογίζονται με βάση έναν αλγόριθμο που έχουμε υλοποιήσει σε Matlab και C που κάνει εξαγωγή των δεδομένων από την SBML διαμέσου της C με το αρχείο που δείχνει τους δείκτες (pointers) για την συμπλήρωση του DG πίνακα και εφαρμόζεται στην Python. Η διαδρομή των δεδομένων αρχικά είναι η ίδια με τον FRM4X που περιγράψαμε προηγουμένως σε τρόπο λειτουργίας 2. Δηλαδή η MMU στέλνει τις κατάλληλες διευθύνσεις στα RTs τα οποία με τη σειρά τους στέλνουν τις διευθύνσεις στην ST και τα (k reaction rates) της κάθε αντίδρασης στα PEs για τον υπολογισμό N χρόνων με βάση και των δεδομένων από την μνήμη (Flag Table). Τα N PEs θα δώσουν τα $\tau_j(x)$ και τις διευθύνσεις στα αντίστοιχα N κομμάτια με τη σειρά για το κάθε N PQ ή FIFO. Μόλις αποθηκευθούν οι χρόνοι στις FIFO ή στα PQs όλα τα δεδομένα θα δοθούν στην αντίστοιχη μονάδα MTU όπου από τα δεδομένα από το κάθε PE θα υπολογιστούν οι νέοι ελάχιστοι χρόνοι και οι αντίστοιχες διευθύνσεις. Κατόπιν με τον ίδιο τρόπο όπως και στον FRM θα υπολογιστούν οι νέες συγκεντρώσεις από τις $N(q+p)$ βαθμίδες SU και οι N χρόνοι από τις N TUs για τον υπολογισμό, της εξέλιξης του στιγμιότυπου χρόνου $t = t_{old} + \tau_{min}$.

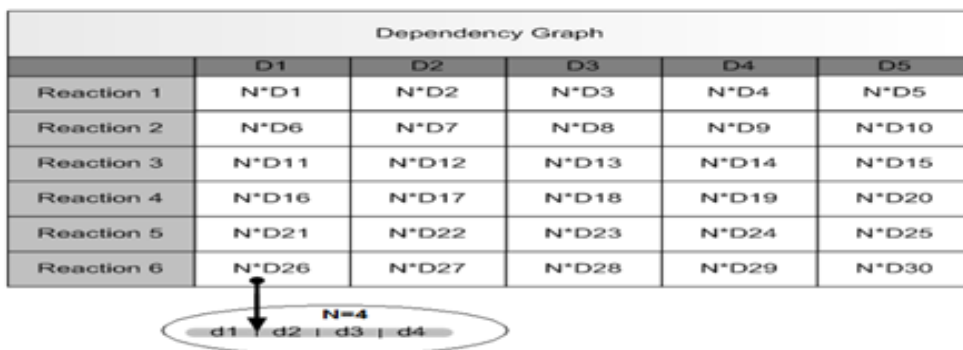
“Επειτα η διεύθυνση που αντιστοιχεί στην αντίδραση που κέρδισε θα περάσει μέσω της MMU στο DG το οποίο με τη σειρά του για την κάθε N αντίδραση θα δώσει N ομάδες από εξαρτήσεις στα N RTs. Το κάθε ένα RT θα ενεργοποιήσει το 2^ο κομμάτι του ST και θα δώσει

τα δεδομένα στα N PEs. Τα δεδομένα από τα PEs με τα $\tau_j(x)$ και $a_j(x)$ που υπολογίστηκαν θα αντιγραφούν στα N κομμάτια είτε της FIFO είτε του PQ τα οποία βασίζονται στις N διευθύνσεις του DG. Από εκεί το κάθε κομμάτι εκ των N PQ θα δώσει με τη σειρά που εκτελούνται τα $\tau_j(x)$ στα N MTU1 από όπου θα υπολογιστεί ο χρόνος της νέας επόμενης αντίδρασης $\tau_{\mu}(x)$ και με αυτόν τον τρόπο θα συνεχισθεί η διαδικασία μέχρι το t να γίνει t_{max} . Για τον NRM και για τους 2 τρόπους λειτουργίας υπάρχει η δυνατότητα πολλαπλών προσομοιώσεων με τις αντίστοιχες κρυφές μνήμες ST όπου βρίσκονται οι αρχικές συγκεντρώσεις. Για την λειτουργία του FRM4X ισχύει η ίδια ανάλυση και για τους 2 τρόπους που περιγράφηκε προηγουμένως στο κεφάλαιο 3.

4.3 Περιγραφή και οργάνωση επιπλέον μνημών για την υλοποίηση του NRM

Στο παρόν κεφάλαιο θα περιγράψουμε την οργάνωση κάποιων επιπλέον μνημών που χρειάζονται για την λειτουργία του αλγορίθμου NRM όπως είναι ο γράφος εξαρτήσεων DG. Όπως αναφέραμε ο γράφος αυτός δίνει τις εξαρτήσεις για την κάθε αντίδραση. Κάθε αντίδραση αντιστοιχίζεται σε μια διεύθυνση, κάθε γραμμή του γράφου περιγράφει ποιές αντιδράσεις επηρεάζονται εάν πραγματοποιηθεί η συγκεκριμένη αντίδραση. Ο μέγιστος αριθμός για αυτήν την σχεδίαση είναι 80 λόγω του μεγέθους της μνήμης. Σαφώς αυτό μπορεί να παραμετροποιηθεί και ο αριθμός να γίνει πιο μεγάλος το οποίο βέβαια σχετίζεται και από το διαθέσιμο FPGA (χωρητικότητα σε block Rams).

Όπως φαίνεται στο σχήμα 39 η μνήμη DG είναι χωρισμένη σε N κομμάτια (για το συγκεκριμένο παράδειγμα $N=4$) και το κάθε κομμάτι αποτελείται από πίνακα μεγέθους M όπου κάθε γραμμή στην περίπτωση μας έχει 10 στήλες των 8 στοιχείων σε hex αριθμούς που υποδηλώνουν τις αντιδράσεις που επηρεάζονται από την πραγματοποίηση της κάθε αντίδρασης. Δηλαδή οι 8 αριθμοί είναι 8 pointers που δείχνουν στους πίνακες RT.



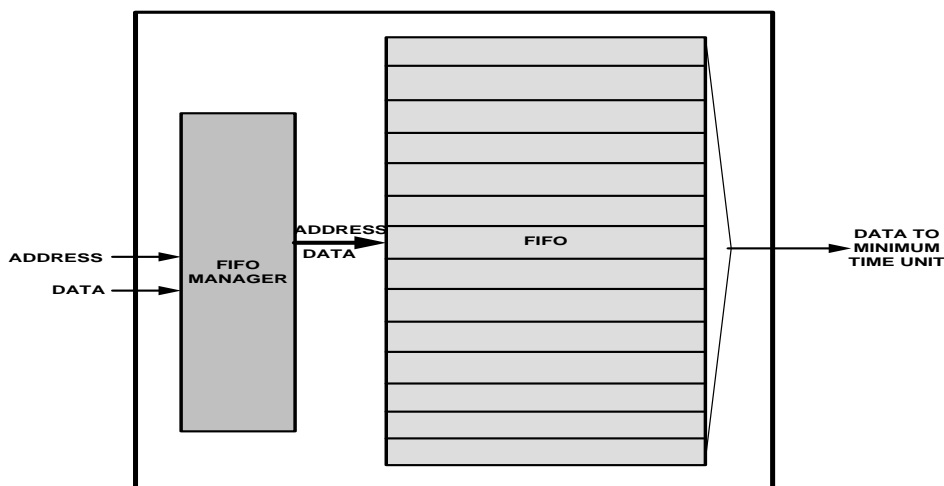
Σχήμα 39: Οργάνωση γράφου εξαρτήσεων

Ο μηχανισμός έχει ως εξής: αρχικά δημιουργείται με βάση έναν αλγόριθμο από το αρχείο της SBML ο πίνακας των αλληλεξαρτήσεων. Από όλες τις αλληλεπιδράσεις αυτές

διαλέγεται ο μέγιστος αριθμός αλληλοεπιδράσεων. Ενημερώνονται εσωτερικά στην MMU κάποιοι μετρητές για αυτόν τον αριθμό και για την κάθε αντίδραση θα αρχίσουν να μετρώνε οι μετρητές μέχρι αυτόν τον μέγιστο αριθμό. Όσες αντιδράσεις έχουν λιγότερες αλληλεξαρτήσεις θα έχουν '0' στις θέσεις αυτές το οποίο θα αναγνωρίζεται από κάποιους εσωτερικούς συγκριτές και θα ενημερώνεται με σήμα διακοπής η μονάδα ελέγχου. Η κάθε στείλει αποτελείται από μία λέξη D περιέχει 8 εξαρτήσεις δηλαδή διευθυνσιοδοτήσεις προς την μνήμη RT που δείχνει τις αντιδράσεις. Σε κάθε κύκλο θα δίδονται τόσες εξαρτήσεις παράλληλα όσα και ο αριθμός των PEs δηλαδή για το σχήμα 39 για $N=4$ θα επεξεργάζονται σε κάθε κύκλο 4 εξαρτήσεις για να υπολογισθούν οι νέες πιθανότητες αντίδρασης και οι νέοι χρόνοι.

4.4 Μέθοδοι ταξινόμησης και εύρεσης ελάχιστης τιμής χρόνου

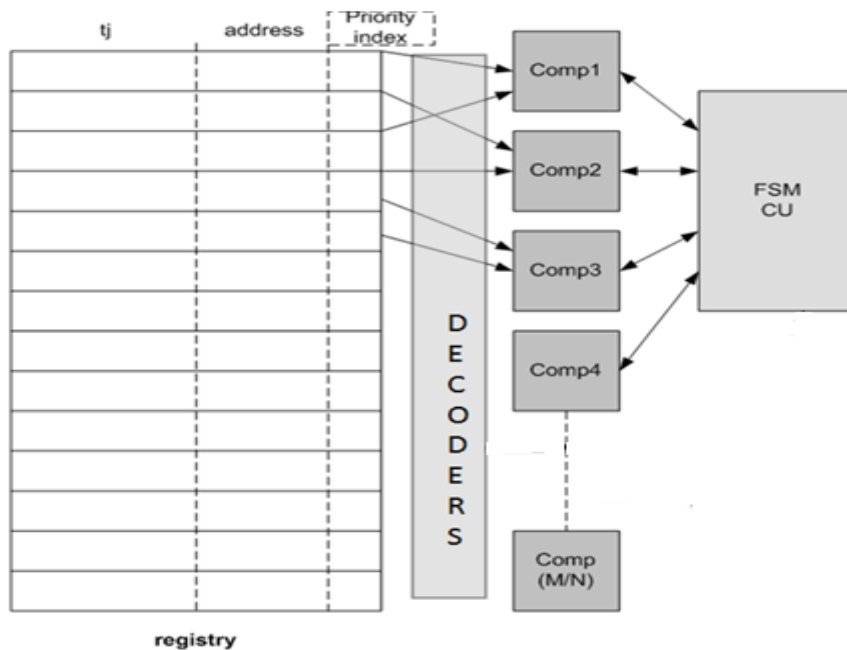
Σε αυτή την ενότητα θα αναλύσουμε όλους τους τρόπους κατηγοριοποίησης και εύρεσης του ελάχιστου χρόνου αντίδρασης που χρησιμοποιήθηκαν ώστε να επιταχυνθεί η διαδικασία του αλγορίθμου. Στο σχήμα 40 φαίνεται η μνήμη κράτησης των δεδομένων που ουσιαστικά κρατάει τα $\tau_j(x)$ και τις αντίστοιχες διευθύνσεις και χωρίζεται σε N τομείς. Ο κάθε τομέας αποτελείται από πίνακες κράτησης των δεδομένων για το κάθε N PE. Τα δεδομένα των υπολογιζόμενων χρόνων και των αντίστοιχων διευθύνσεων αποθηκεύονται σε συγκεκριμένες χαρακτηριστικές θέσεις με βάση τη σειρά τους στο δίκτυωμα των αντιδράσεων (reaction index). Σε όσες αντιδράσεις υπάρχει αλλαγή στις συγκεντρώσεις των στοιχείων τους για αυτές θα πρέπει να υπολογιστούν οι χρόνοι αντίδρασης και αυτοί οι νέοι χρόνοι θα αντικαταστήσουν τους παλιούς στους πίνακες κράτησης δεδομένων άρα και θα αποθηκευθούν στις ίδιες θέσεις. Από εκεί τα δεδομένα θα περάσουν στη δομή των MTUs για να γίνουν οι συγκρίσεις και να βρεθούν οι ελάχιστοι ή ο ελάχιστος χρόνος. Άρα ουσιαστικά σε αυτήν την μνήμη αποθηκεύονται και θα συγκριθούν τα δεδομένα των αντιδράσεων που επηρεάστηκαν με τα δεδομένα που δεν επηρεάστηκαν για να βρεθεί τελικά ο ελάχιστος αυτός χρόνος.



Σχήμα 40: Περιγραφή Μνημών FIFO για κράτηση των δεδομένων.

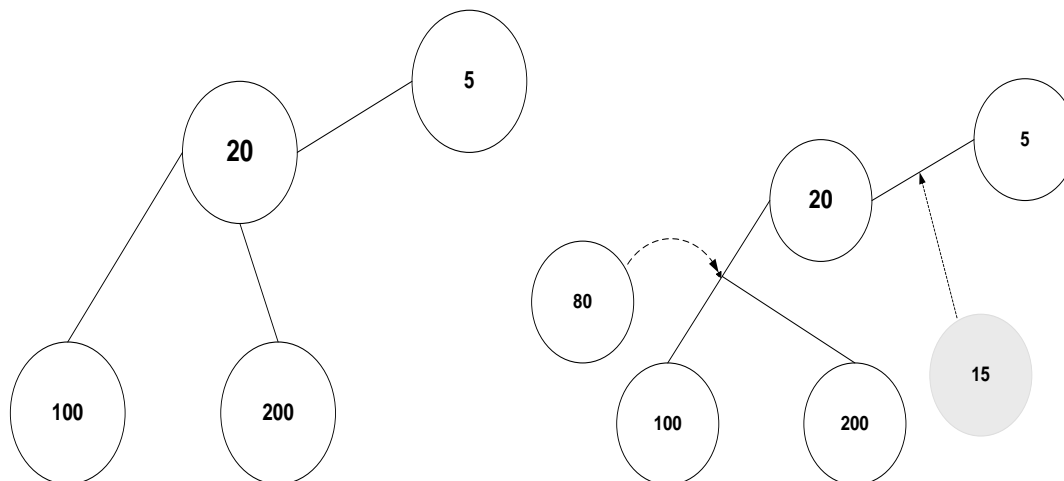
Επιπροσθέτως για να επιταχύνουμε επιπλέον τις διαδικασίες εύρεσης του μικρότερου χρόνου που αποτελεί τελικά το κρίσιμο μονοπάτι της μεθόδου NRM κατασκευάσαμε 2 μεθόδους ταξινόμησης των δεδομένων χρησιμοποιώντας δένδρα προτεραιότητας και πολλαπλούς παράλληλους ταξινομητές. Αρχίζοντας από τους N παράλληλους ταξινομητές, ο κάθε ταξινομητής αποτελείται από $2(M/N - 1)$ συγκριτές και καταμετρητές, $N * M/N$ πίνακες κράτησης δεδομένων και N βαθμίδες αποκωδικοποιητών όπου στέλνουν χαρακτηριστικά σήματα σε μία τοπική μονάδα ελέγχου. Οι πίνακες κράτησης δεδομένων έχουν την μορφή που φαίνεται Στο σχήμα 41 και περιέχουν τους χρόνους και τις αντίστοιχες διευθύνσεις τους που δείχνουν τις αντίστοιχες αντιδράσεις καθώς επίσης και περιέχουν κάποιους δείκτες (priority indexes) οι οποίοι δείχνουν την σειρά ταξινόμησης και κράτησης των δεδομένων στους πίνακες.

Τα δεδομένα των χρόνων φυλάσσονται πάντα στις ίδιες θέσεις και γίνεται σύγκριση των τιμών αναμεταξύ τους από τους $2(M/N-1)$ συγκριτές και ανάλογα με το αποτέλεσμα σύγκρισης ενεργοποιούνται κάποια σήματα τα οποία μέσω των N αποκωδικοποιητών και της τοπικής μονάδας ελέγχου είτε αυξάνουν είτε μειώνουν τους καταμετρητές που δείχνουν τη θέση στον πίνακα δεδομένων. Αυτή η διαδικασία συνεχίζεται έως ότου όλα τα δεδομένα βρεθούν κατηγοριοποιημένα. Η κάθε κατηγοριοποίηση έχει 6 κύκλους καθυστερήσεις και πολυπλοκότητα $O(\log M/N)$.



Σχήμα 41: Περιγραφή τρόπου πολλαπλών ταξινόμησεων.

Η 3^η μέθοδος είναι η κατασκευή ενός συστήματος δένδρων προτεραιότητας. Τα δεδομένα όπως και στην προηγούμενη μέθοδο φυλάσσονται σε συγκεκριμένες θέσεις χαρακτηριστικές της αντίδρασης από την οποία προέρχονται μαζί με την αντίστοιχη διεύθυνση. Αναλόγως του αριθμού των χρονικών στιγμών κτίζεται ένα δένδρο με βάση το αποτέλεσμα της σύγκρισης όπου πάλι δεν γίνεται αντιμετάθεση των δεδομένων απλά ενημερώνεται ο δείκτης (priority index) και δείχνει την θέση του στοιχείου στον πίνακα. Πάντα στο επάνω κομμάτι του δένδρου βρίσκεται η μικρότερη τιμή και αντιστοίχως όλες οι άλλες. Το παρακάτω σχήμα δείχνει την δημιουργία των δένδρων.



Σχήμα 42: Περιγραφή μηνύων για οργάνωση και αποθήκευση των υπολογιζόμενων δεδομένων σε δένδρα προτεραιότητας και για τους 2 τρόπους λειτουργίας.

Η μορφή και η παραλληλοποίηση των συγκριτών γίνεται από την διαδικασία της σύνθεσης του κυκλώματος μέσω του προγράμματος SysPy. Η υλοποίηση του FRM4X_NRM4X προσφέρει και την δυνατότητα σύγκρισης των αλγορίθμων ανάλογα με το βιοχημικό μοντέλο για την εξαγωγή συμπερασμάτων. Επίσης εκεί που μια NRM σχεδίαση μπορεί να μην μπορεί να ανταπεξέλθει όπως στην περίπτωση που οι εξαρτήσεις είναι πάρα πολλές ή και ταυτόσημες με το σύνολο των αντιδράσεων ο FRM μπορεί να είναι πιο αποδοτικός και άρα να προγραμματιστεί το σύστημα σε τρόπο λειτουργίας FRM4X. Επίσης και σε αυτό το SoC μπορεί να γίνει αντίστοιχη στατιστική επεξεργασία των δεδομένων από την μονάδα ST-block και για τους 2 τρόπους λειτουργίας και στους 2 αλγορίθμους.

4.5 Στατιστικά στοιχεία προσομοίωσης για την υλοποίηση του FRM4X_NRM4X αλγορίθμου

Για το σύστημα του FRM4X και NRM4X που προσομοιώνει και τους 2 αλγορίθμους δίνονται τα παρακάτω στατιστικά στοιχεία για την σχεδίαση σε Xilinx. Παρατηρούμαι ότι η μέγιστη συχνότητα λειτουργίας που παρατηρήθηκε ανήλθε στα 266 MHz με κύκλους καθυστερήσεων Latency, που είναι ανάλογοι του βαθμού της αντίδρασης και κυμαίνονται ξεχωριστά για τον FRM αλγόριθμο με μέγιστη καθυστέρηση τους 83 κύκλους για μέχρι 3^{ου} βαθμού αντιδράσεις. Οι 70 κύκλοι αντιπροσωπεύουν την χειρότερη περίπτωση εκ των 6 του Πίνακα 1 του κεφαλαίου 2 όπου είναι η περίπτωση 3^{ου} βαθμού αντίδραση με όλα τα στοιχεία ίδια, καθώς επίσης και για τον NRM με μέγιστη καθυστέρηση $83 + M/N * \log(M/N)$ (επιπλέον καθυστέρηση λόγω των ταξινομητών). Η συχνότητα λειτουργίας του SoC FRM_NRM4X σε σχέση με την FRM4X υλοποίηση διαπιστώθηκε ότι πέφτει λόγω της αυξανόμενης πολυπλοκότητας του συστήματος καθώς προστέθηκαν επιπλέον υπομονάδες στα συστήματα μνημών όπως πολυπλέκτες με αποτέλεσμα στον κύριο όγκο ανταλλαγής δεδομένων με τις μνήμες να υπεισέρχονται επιπλέον καθυστερήσεις.

Πίνακας 9: Στατιστικά στοιχεία P.P.R προσομοίωσης για επιδόσεις M-reactions/sec,καθυστερήσεις και απόδοση λειτουργίας για την FRM4X-NRM4X SoC σχεδίαση σε Xilinx Virtex 5-7 FPGAs.

Xilinx (Virtex 5 XC5VLX110T -155T-1ff1136 – Virtex 7 XC7V855T-3ffg1157)						
Component	Frequency	Latency (cycles)	LUTS	DSPs	Regs	RAM
FRM4X-NRM4X	200*-266**	(44-70) + 13/6*log(N)/Nlog(N)	53419-68750	64	64435-74369	116/120

*Μέγιστη συχνότητα λειτουργίας για το FPGA Virtex 5.

** Μέγιστη συχνότητα λειτουργίας για το FPGA Virtex 7.

Στον πίνακα φαίνονται όλοι οι κύκλοι καθυστέρησης για τις διάφορες μεθόδους ταξινόμησης που χρησιμοποιούνται.

Η σχεδίαση του FRM4X-NRM4X SoC μπορεί να προσομοιώσει για βιοχημικά συστήματα μέχρι 3^{ου} βαθμού αντιδράσεων και μέχρι 8 στοιχεία σε κάθε αντίδραση (3 αντιδρώντα και 5 προϊόντα) καθώς επίσης υπάρχει και δυνατότητα υποστήριξης και των 2 τρόπων λειτουργίας. Ο πίνακας 9 παραθέτει στατιστικά στοιχεία για την σχεδίαση με τους NRM-FRM αλγόριθμους παραλληλοποιημένους με 4 επεξεργαστές καθώς επίσης φαίνεται και την απόδοση σε M-reaction/ sec. Με βάση την ανάλυση του κεφαλαίου 3 κατά τη 1^η προσομοίωση (1st run) του NRM αλγορίθμου που ουσιαστικά έχω υλοποίηση του FRM, στην αρχή παρατηρούνται περίπου οι ίδιες καθυστερήσεις. Έπειτα όμως λόγω του γράφου εξαρτήσεων παρατηρούμαι ότι οι εσωτερικές καθυστερήσεις εξαρτώνται κυρίως από τις εξαρτήσεις τις κάθε αντίδρασης στο δίκτυο των αντιδράσεων του Βιοχημικού Μοντέλου καθώς και του τρόπου ταξινόμησης. Αυτός είναι και ο κύριος λόγος αύξησης της απόδοσης του NRM σε σχέση με τον FRM αλγόριθμο. Όπως παρατηρείται και από τον πίνακα 9 το κρίσιμο μονοπάτι έγκειται στον υπολογισμό του μικρότερου χρόνου και στην ενημέρωση των PQ για αυτό και όπως είδαμε πραγματοποιήθηκαν 3 διαφορετικές σχεδιάσεις για την εύρεση της βέλτιστης λύσεις.

4.6 Μονάδα στατικών στοιχείων

Για την αύξηση της απόδοσης του συστήματος προσομοίωσης σχεδιάστηκε εσωτερικά μία μονάδα ανάκτησης στατιστικών στοιχείων (St-block). Η στοχαστική προσομοίωση από την φύση της στηρίζεται στην επανάληψη των προσομοιώσεων και την εύρεση του μέσου όρου των συγκεντρώσεων από την κάθε επανάληψη της διαδικασίας. Η μέθοδος αυτή αποδείχθηκε αρκετά χρονοβόρα όταν γίνεται στο τερματικό (Matlab). Για το λόγο αυτό υλοποιήθηκε αυτή η μονάδα όπου σε συνεργασία με την DDR2 μνήμη του FPGA και έναν μηχανισμό ανάκτησης των δεδομένων να μπορεί να υπολογιστεί ο μέσος όρος συγκέντρωσης για τα στοιχεία ενδιαφέροντος της βιοχημικής αντίδρασης.

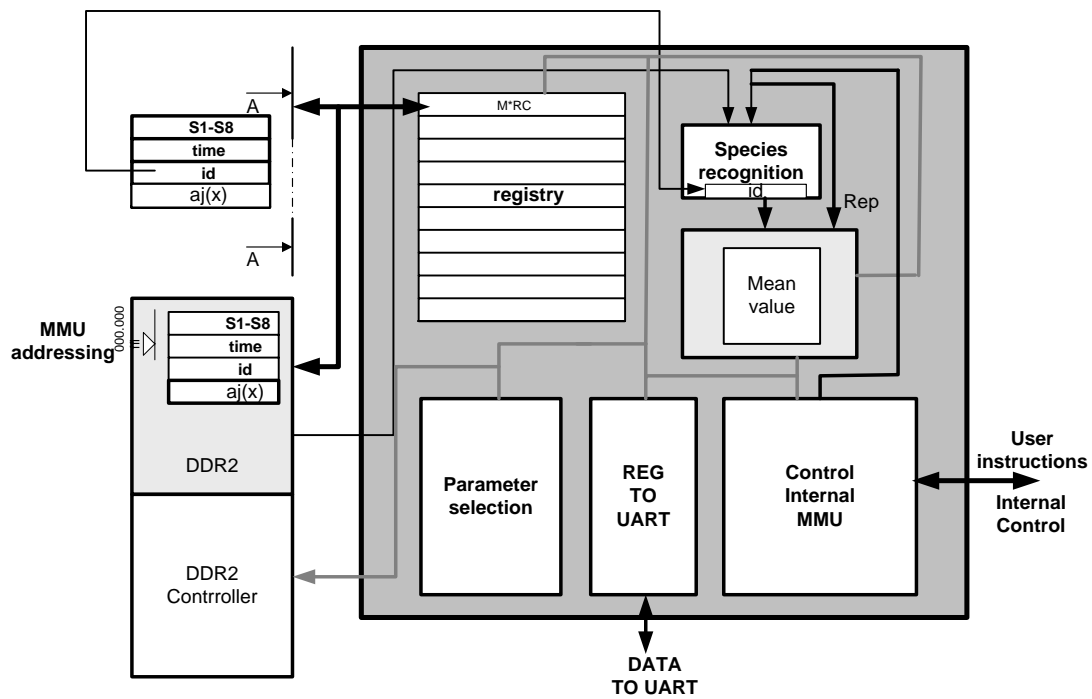
Από τον χρήστη επιλέγονται ο χρόνος προσομοίωσης, και ο αριθμός των επαναλήψεων της κάθε προσομοίωσης. Επίσης καταχωρούνται από τον χρήστη το Reaction index (Rid)

που δείχνει από ποια αντίδραση θέλουμε να επεξεργαστούμε τα στοιχεία και το Sid που δείχνει ποια στοιχεία θα επιλέξουμε που απαρτίζουν την αντίδραση αυτή. Αυτές οι μεταβλητές κρατιούνται στην τοπική μονάδα ελέγχου της μονάδος.

Εσωτερικοί απαριθμητές καταμετρούν μετά από κάθε κύκλο αντίδρασης πόσοι κύκλοι αντίδρασης έγιναν. Αναλόγως του μεγέθους των πακέτων και του αριθμού των προσομοιώσεων γίνεται το χώρισμα της DDR2 με βάση την βαθμίδα διευθυνσιοδότησης. Εάν υπερβούν οι απαιτήσεις σε μνήμη ενημερώνεται ένα εσωτερικό σήμα διακοπής (Interrupt) το οποίο ειδοποιεί τον χρήστη ανάβοντας το LED 4 στο FPGA. Ανάλογα ο χρήστης πρέπει να μειώσει τα στοιχεία προς επεξεργασία.

Τα δεδομένα για την κάθε προσομοίωση αποθηκεύονται στην εσωτερική DDR2 σε μορφή πακέτων όπου το κάθε πακέτο περιλαμβάνει 8 στοιχεία, το χρόνο και το id της αντίδρασης που κέρδισε στο κύκλο αντίδρασης. Εάν θέλουμε 10 διαφορετικές προσομοιώσεις των 4k πακέτων τότε θα έχω 40 K δεδομένων στη DDR2. Από αυτά ο χρήστης ενημερώνετε για το στοιχείο που θέλει να δει εκ των 8 και τις αντιδράσεις στις οποίες συμμετέχει. Ένας αποκωδικοποιητής συγκεντρώνει τα στοιχεία ενδιαφέροντος με βάση τις μεταβλητές Sid και Rid από την DDR2 από την κάθε επανάληψη και αθροίζει επαναλαμβανόμενα τα στοιχεία ενδιαφέροντος για το κάθε στιγμιότυπο χρόνου και τα διαιρεί διά των αριθμό των επαναλήψεων προσομοίωσης.

Με αυτόν τον μηχανισμό που φαίνεται και στο σχήμα 43 υπολογίζονται οι μέσοι όροι των στοιχείων ενδιαφέροντος. Πιο αναλυτικά έχουν σχεδιαστεί απαριθμητές οι οποίοι καταμετρούν ανάλογα με τον αριθμό των επαναλήψεων για να γίνουν συγκεκριμένες αθροίσεις των στοιχείων και έπειτα να διαιρεθούν με τον αριθμό των επαναλήψεων ώστε να υπολογιστούν οι μέσοι όροι. Όλες οι μονάδες είναι παραμετροποιημένες και ανάλογα με το διαθέσιμο αναπτυξιακό ο χρήστης μπορεί να παραλληλοποιήσει πλήρως την λειτουργία της μονάδος. Έτσι ανάλογα με τον τρόπο λειτουργίας και τον αριθμό των στοιχείων προς επεξεργασία και δεδομένου της τεχνολογίας του FPGA μπορεί να υπάρχουν $N*(q+p)$ DSP μονάδες (Mean Units) καθώς και N Species Recognition (σχήμα 43) ενότητες και N εσωτερικές μνήμες για την κράτηση των δεδομένων. Ενδιαφέρον παρουσιάζει το γεγονός ότι η διαδικασία και το υλικό που έχει κατασκευαστεί μπορεί να χρησιμοποιηθεί από τους ODM (Optimized Direct Method)[43] και wNRM αλγόριθμο[43] για την εύρεση των αντιδράσεων που εμφανίζονται πιο συχνά και των αντιδράσεων που επηρεάζουν πιο πολύ την ισορροπία του συστήματος



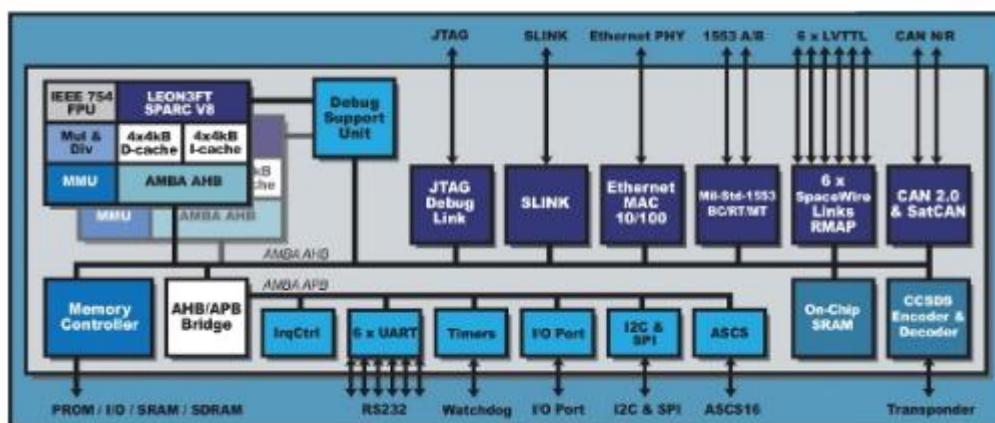
Σχήμα 43: Περιγραφή λειτουργίας St-block.

ΚΕΦΑΛΑΙΟ 5

Αυτοματοποίηση Διαδικασιών

5.1 Εισαγωγή του επεξεργαστή LEON στα SoCs και διασύνδεση και προγραμματισμός μέσω του SysPy

Σε αυτό το κεφάλαιο θα μιλήσουμε για την συνένωση όλων των σχεδιάσεων FRMxX ή FRM4X_NRM4X με τον επεξεργαστή Leon και του προγράμματος SysPy. Με αυτόν τον τρόπο δημιουργήσαμε ένα περιβάλλον ώστε να μπορεί ο χρήστης να επεξεργάζεται καλύτερα τα δεδομένα και να μπορεί πολύ εύκολα να συνθέτει κυκλώματα ανάλογα με το εκάστωτε μοντέλο χωρίς προαπαιτούμενες γνώσεις σε VHDL. Πριν ξεκινήσουμε την ανάλυση μας θα κάνουμε μια γρήγορη προεπισκόπηση στον επεξεργαστή Leon του οποίου το Block διάγραμμα φαίνεται παρακάτω (σχήμα 44).

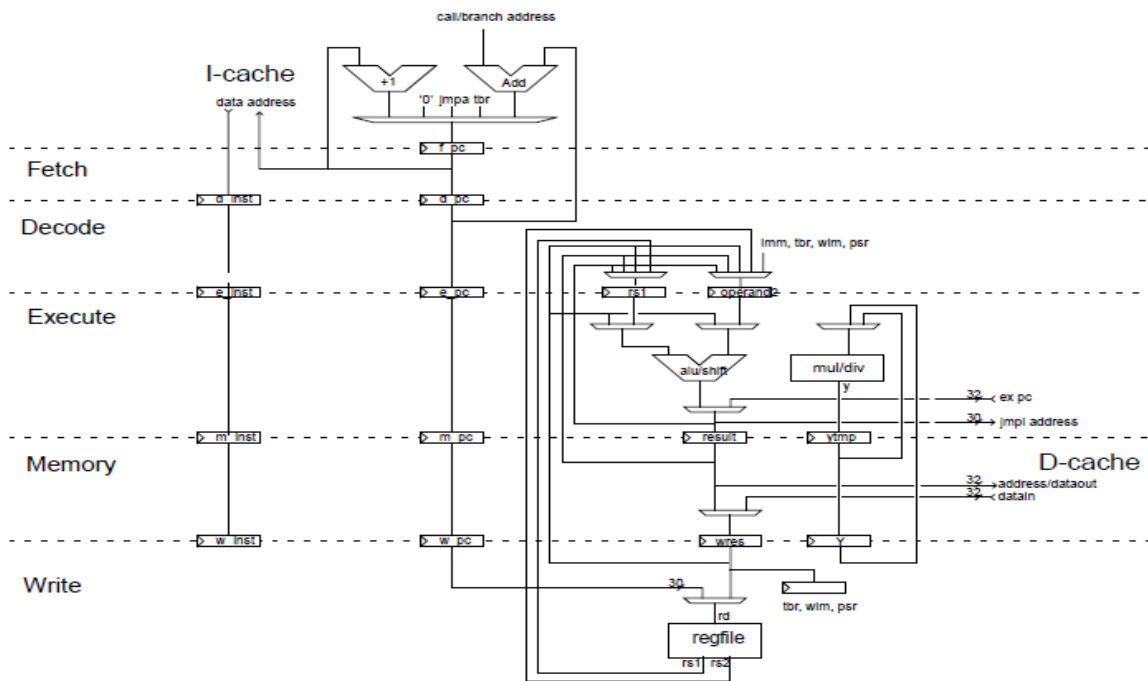


Σχήμα 44 :Διάγραμμα επεξεργαστή Leon.

Ο επεξεργαστής Leon[14] δημιουργήθηκε στα πλαίσια μιας συνεργασίας της ESA (Ευρωπαϊκή εταιρεία Διαστήματος) με την εταιρεία Gaisler για την δημιουργία ενός παραμετρικού αναδομήσιμου (parametric, reconfigurable) επεξεργαστή με χρησιμότητα σε αεροδιαστημικές εφαρμογές. Πρόκειται για έναν ενσωματωμένο επεξεργαστή που έχει σχεδιαστεί σε VHDL και μπορεί να κατέβει σε διάφορα FPGAs όπως Altera Xilinx Achronix κτλπ, ο οποίος διαθέτει ένα γραφικό περιβάλλον (GUI interface) με το οποίο ο χρήστης, εύκολα μέσω Linux ή Cygwin μπορεί να συνθέσει σε όποιο FPGA διαθέτει τον επεξεργαστή. Ο επεξεργαστής αυτός μπορεί να προγραμματιστεί σε C μέσω του GRMON, ενός προγράμματος που κάνει disassembly του C προγράμματος και ουσιαστικά επικοινωνεί με το FPGA είτε μέσω σειριακής θύρας είτε καλωδίου JTAG και προγραμματίζει τον επεξεργαστή. Ο επεξεργαστής αυτός διαθέτει πολλές δυνατότητες (χαρακτηριστικό είναι ότι μπορεί να λειτουργήσει με Linux σαν operating system και να τρέξει κάποιες εφαρμογές) επίσης είναι open core δηλαδή μπορεί ο καθένας να τον

κατεβάσει και να τον χρησιμοποιήσει.

Συνοπτικά ο επεξεργαστής διαθέτει μία κύρια μονάδα (Integer Unit) σε αρχιτεκτονική βασισμένη στην Spark V8 όπως φαίνεται στο σχήμα 45 η οποία γενικά διαθέτει 5 στάδια σωλήνωσης ,έχει 2 register files των 32 , έχει αναδομήσιμους πολλαπλασιαστές (16x16, 32x8, 32x16 & 32x32),16x16 bit Multiply Accumulator των 40-bit και μονάδα διαίρεσης (Radix-2 divider (non-restoring)) και φυσικά μονάδα ελέγχου και κρυφές μνήμες (Data Cache και Instruction Cache). Επάνω σε αυτήν την μονάδα και μέσω διαύλου μεταφοράς των δεδομένων (AMBA AHB/APB buses) τοποθετούνται όλες οι άλλες περιφερειακές μονάδες όπως DDR2, UART controllers, RAMS, ROMS,VGA controller, SPI Controller, Ethernet Controller, JTAG controller, Flash memory controller καθώς υπάρχουν επίσης και άλλα interfaces για πιο εξειδικευμένες εφαρμογές οι οποίες μονάδες δεν είναι δωρεάν και απαιτείται ειδική άδεια όπως ειδική FPU μονάδα για DSP εφαρμογές όπου απαιτούνται αριθμητικές πράξεις κινητής υποδιαστολής.



Σχήμα 45:Διάγραμμα βασικής κύριας μονάδας του Leon.

Ο δίαυλος δεδομένων AMBA λειτουργεί με μια λογική αφέντη –σκλάβου (Master –slave). Οτιδήποτε εφαρμοστεί και θέλει να επικοινωνήσει με το Leon θα πρέπει να συνδεθεί με το AMBA και να λειτουργεί σαν σκλάβος. Επίσης επάνω στον δίαυλο δεδομένων AMBA υπάρχει ειδική μονάδα GPIO του Leon από την οποία οποιαδήποτε σύστημα εξωτερικό μπορεί να συνδεθεί και να ανταλλάχτουν τα δεδομένα με το Leon.

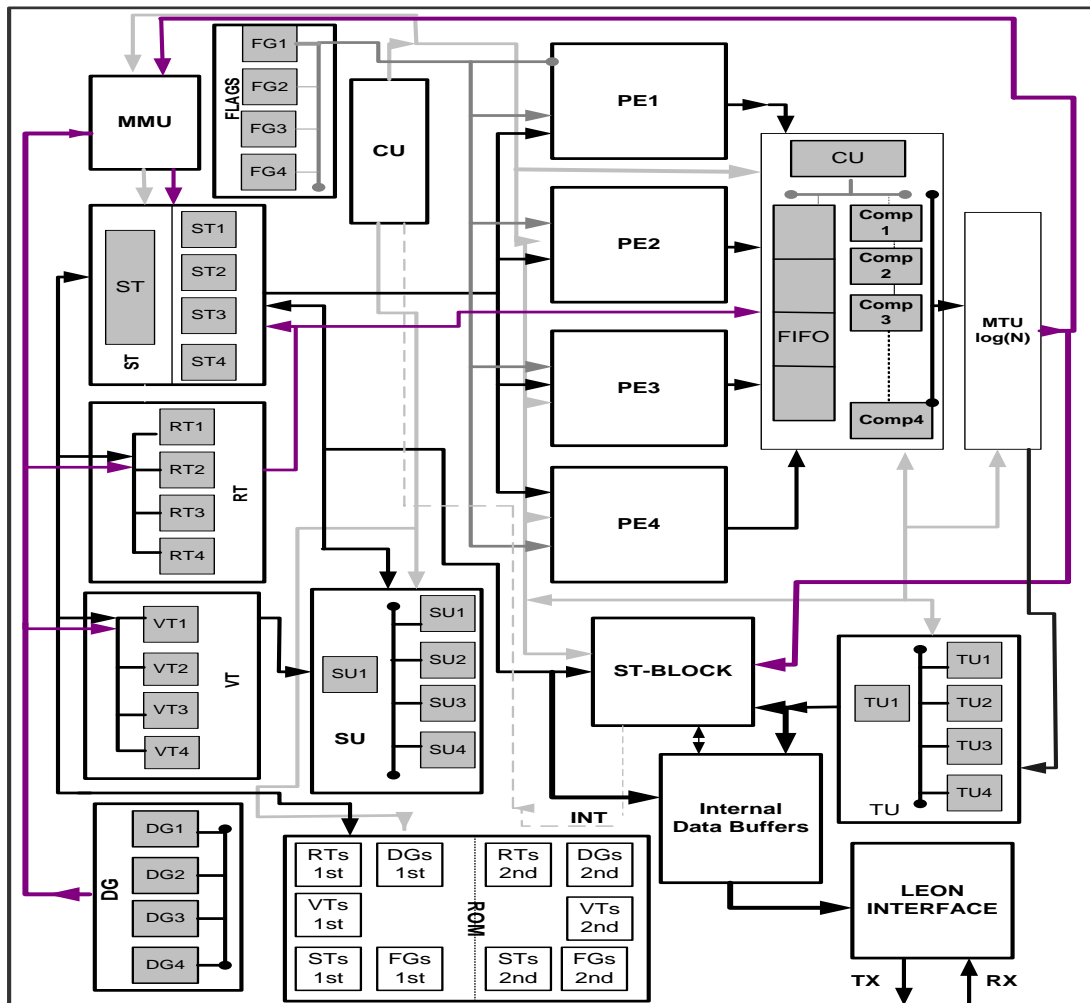
5.2 Ένωση των SoCs με το AMBA Bus του Leon

Για να ενώσουμε τα SoCs με τον επεξεργαστή Leon σχεδιάστηκε ένα σύστημα ελέγχου και αποθήκευσης δεδομένων για την ανταλλαγή δεδομένων με τα GPIO ports. Οπότε το σύστημα των FRMxX SoCs ή FRM4X_NRM4X λειτουργεί σαν σκλάβος αφού ενώνεται διαμέσου των GPIOs θυρών στο AMBA bus.

Το σύστημα λοιπόν αυτό διαθέτει μια κεντρική FSM η οποία ανταλλάσει σήματα τόσο με τον επεξεργαστή Leon όσο και από το σύστημα των SoCs. Επίσης υπάρχει ένα σύστημα από μνήμες διπλής και τετραπλής μεταφοράς δεδομένων (dual,quad buffering) τα οποία μέσω μια ειδικής σχεδιασμένης μονάδας ελέγχου που επιπλέον επικοινωνεί και συνεργάζεται με την κεντρική μονάδα ελέγχου ανταλλάσει σήματα με τον επεξεργαστή Leon τα οποία αποστέλλονται σε μορφή πακέτων των 10 στοιχείων των 32 bit.

Ο επεξεργαστής Leon έχει προγραμματιστεί για να δέχεται τα πακέτα και να τα αποθηκεύει στην εσωτερική μνήμη DDR2 του FPGA από όπου στέλνονται μέσω ενός Ethernet controller που ελέγχεται από το Leon και επικοινωνεί με την Ethernet θύρα του FPGA ώστε από εκεί να αποσταλούν στην Ethernet θύρα του Η/Υ. Από εκεί τα δεδομένα διαμέσου Python script θα μετατραπούν κατάλληλα και θα δημιουργήσουν αυτόματα ένα Matlab αρχείο. Όπου μέσω προγράμματος γραμμένου σε Matlab θα γίνει η επεξεργασία των δεδομένων και θα σχηματισθούν οι κυματομορφές.

Το πακέτο των δεδομένων που στέλνεται από τα SoCs στο Leon αποτελείται από 10 στοιχεία των 32 bits και είναι ο χρόνος προσομοίωσης, οι συγκεντρώσεις των στοιχείων που αλλάζουν (για 8 στοιχεία 3 αντιδρώντα και 5 προϊόντα) καθώς επίσης δηλώνεται η αντίδραση που γίνεται κάθε φορά (reaction index). Ο μηχανισμός αυτός φαίνεται και από την σχήμα 46 και πιο χαρακτηριστικά από τις μονάδες των memory blocks και της μονάδας Leon Interface.



Σχήμα 46: Διάγραμμα διασύνδεσης FRM4X_NRM4X με επεξεργαστή Leon.

Με αυτόν τον τρόπο μπορούμε να δούμε την αντίδραση που πραγματοποιείται κάθε χρονική στιγμή και την μεταβολή των συγκεντρώσεων στα στοιχεία της αντίδρασης αυτής. Στο Matlab με βάση το index της αντίδρασης και το αρχείο από τους ενδείκτες pointers καθώς και του αρχείου με τις αρχικές συγκεντρώσεις που έχει παρθεί μέσω κώδικα σε C από την SBML, και με βάση τις μεταβολές των συγκεντρώσεων αναδημιουργείται ένας πίνακας με όλα τα στοιχεία του μοντέλου για την κάθε επανάληψη του αλγορίθμου. Από όλες τις επαναλήψεις που εντοπίζονται υπολογίζονται οι μέσοι όροι και από αυτά με βάση το στιγμιότυπο χρόνου δημιουργούνται οι εκάστωτε κυματομορφές για τα στοιχεία που συμμετέχουν στις βιοχημικές αντιδράσεις. Οι κυματομορφές συγκρίνονται έπειτα με τους αντίστοιχους Software προσομοιωτές Copasi και Ibiosim tool.

5.3 Σύστημα επικοινωνίας χρήστη

Εκτός από τον επεξεργαστή Leon και της επικοινωνίας μέσω του SysPy έχουν αναπτυχθεί και άλλα συστήματα για επικοινωνία όχι όμως τόσο αποδοτικά. Αρχικά χρησιμοποιήθηκαν για την διεξαγωγή αποτελεσμάτων DDR2/3 controllers και UART

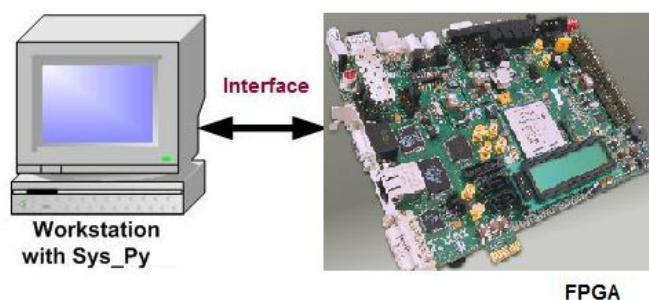
controller για σειριακή επικοινωνία. Για την DDR2/3 επικοινωνία είχαν αναπτυχθεί ειδικές FIFO και μονάδα ελέγχου control για την μεταφορά των δεδομένων σε πακέτα, όπου τα πακέτα από την DDR/32 εμφανίζονταν μέσω σειριακής στο MATLAB. Επίσης για την σειριακή επικοινωνία είχε σχεδιαστεί ένα σύστημα από μνήμες όπου για το κάθε στοιχείο (species), του χρόνου προσομοίωσης και του index κρατούνται δεδομένα των 4K από όπου ο χρήστης μέσω εντολών που έδινε μέσω Matlab μπορούσε να διακρίνει ανά πάσα στιγμή τα στοιχεία ενδιαφέροντος. Επίσης οι μνήμες με βάση ένα interrupt ανανέωναν ανά τακτά χρονικά διαστήματα τα δεδομένα τους. Επιπλέον είχε αναπτυχθεί και Ethernet controller για επικοινωνία και ανταλλαγή δεδομένων απευθείας από το FPGA αλλά η επικοινωνία αποδείχθηκε προβληματική καθώς ο controller μαζί με το σύστημα DDR2 καταλάμβανε πολύ χώρο στο FPGA .

Εξετάζοντας όλα αυτά τα συστήματα αποδείχθηκε ότι η μεγαλύτερη δυσκολία είναι η υλοποίηση και η εύρεση ενός συστήματος επικοινωνίας. Σχεδόν όλα τα συστήματα επικοινωνίας στο FPGA στην ανταλλαγή δεδομένων με τον Η/Υ εισάγουν πολλές καθυστερήσεις. Αυτή η διαδικασία δημιουργεί προβλήματα όταν απαιτούνται γρήγορες ταχύτητες. Όμως καθώς η τεχνολογία των FPGA αναπτύσσεται και έχοντας πετύχει συχνότητες στα 1GHz όπως στο FPGA της κατηγορίας Virtex 7 φαίνεται ότι σε κάποιο σημείο τα συστήματα επικοινωνίας και ανταλλαγής δεδομένων με το Η/Υ θα συγχρονιστούν και δεν θα υπάρχουν πλέον σημαντικές καθυστερήσεις.

Το κομμάτι της επικοινωνίας με τον χρήστη το διαχειρίζεται το πρόγραμμα SysPy. Με βάση κάποιων ειδικών προγραμμάτων (scripts) και μιας διαδικασίας μέσω Python ο χρήστης μέσω ενός εύχρηστου γραφικού περιβάλλοντος σε Linux ή Cygwin μπορεί να προγραμματίσει τους προσομοιωτές. Ο χρήστης θα χρειαστεί να έχει ένα αναπτυξιακό τύπου Virtex 5 καθώς επίσης και να έχει εγκαταστήσει τα προγράμματα της Xilinx και τα πακέτα της Python. Έπειτα ο χρήστης μπορεί να επεξεργαστεί οποιοδήποτε SBML μοντέλο το οποίο περιγράφει ένα βιοχημικό μοντέλο από την SBML βιβλιοθήκη. Μόλις ενεργοποιηθούν τα scripts σε Python θα εμφανιστεί ένα γραφικό περιβάλλον στο οποίο θα επικαλεσθεί να φορτώσει τιμές σε κάποιες μεταβλητές. Όπως να γράψει το τίτλο του μοντέλου, τον αλγόριθμο που θέλει ,τον αριθμό των PEs και άρα επιλέγεται είτε το FRM1X, είτε το FRM2X ή το FRM4X και για μεγαλύτερα FPGA των κατηγοριών Virtex 6-7 υπάρχει η δυνατότητα σύνθεσης και συστήματος FRM8X. Επιπλέον υπάρχει και η δυνατότητα επιλογής FRM4x_NRM4x. Ακόμα επιλέγεται ο τρόπος λειτουργίας βάζοντας μια τιμή σε μια μεταβλητή όπου για Algorithm='0' θα έχω 1 N παράλληλη υλοποίηση και με '1' αντίστοιχα θα έχω N ανεξάρτητες. Έπειτα ρυθμίζεται ο χρόνος προσομοίωσης με βάση

την τιμή της μεταβλητής t_{max} και το εύρος των πακέτων που θα παραλάβει από την Ethernet.

Το σύστημα SysPy πρόκειται για ένα ερευνητικό πρόγραμμα το οποίο έχει γραφτεί σε Python και συνθέτει VHDL κώδικα για την δημιουργία κατάλληλου ψηφιακού συστήματος προσαρμοσμένο στις απαιτήσεις κάποιου προβλήματος. Για την συγκεκριμένη εφαρμογή το σύστημα SysPy δημιουργεί και αρχικοποιεί όλες τις μνήμες των SSA SoC ανάλογα με το SBML μοντέλο που χρησιμοποιείτε και με βάση τις γενικές μεταβλητές που εισάγει ο χρήστης για τον προγραμματισμό της πλατφόρμας, διαλέγει και συνθέτει από μια αυτόματη διαδικασία τον κατάλληλο κώδικα VHDL για ένα σύστημα FRM x X +Leon όπου $x=1,2,4-8$ (αριθμός PE) ή FRM4X_NRM4X+Leon κατάλληλο για την εφαρμογή που έχει επιλεγεί από το χρήστη. Επίσης το SysPy μέσω συναρτήσεων (functions) σε C τις οποίες καλεί, προγραμματίζει αυτομάτως και τον επεξεργαστή Leon. Ακόμα το ίδιο το σύστημα SysPy συνεισφέρει στη διαδικασία ανάκτησης δεδομένων από την Ethernet θύρα του FPGA και στην δημιουργία κατάλληλου αρχείου Matlab για την επεξεργασία των δεδομένων.



Σχήμα 47: Διάγραμμα συστήματος FPGA -SysPy.

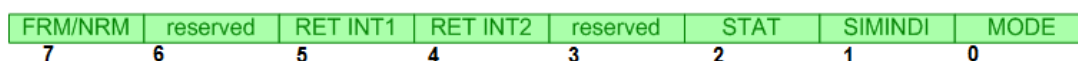
Ουσιαστικά το SysPy είναι ένας μετατροπέας σαν τη system C ο οποίος όμως μπορεί εύκολα να δημιουργήσει ένα εύκολο και κατανοητό περιβάλλον για το χρήστη ώστε χωρίς κώδικα να μπορεί ο χρήστης με βάση τις εκάστως ανάγκες του να δημιουργεί VHDL κώδικα και να συνθέσει hardware. Το SysPy πρόγραμμα αυτοματοποίησε την όλη διαδικασία σύνθεσης υλικού hardware βασισμένο σε κάποιο βιολογικό μοντέλο και πλέον οποιαδήποτε μοντέλο μπορεί εύκολα να προσομοιωθεί.

5.4 Προγραμματισμός συστήματος SysPy με Leon και SSA SoCs

Ο προγραμματισμός της πλατφόρμας γίνεται διαμέσου γενικών παραμέτρων (generic parameters) που εισάγονται στην αρχιτεκτονική (top level). Ο πίνακας 10 δείχνει αναλυτικά αυτές τις παραμέτρους. Οι μεταβλητές αυτές ενημερώνονται διαμέσου της Python και αντανακλούν εσωτερικά σε μεταβλητές για τις διάφορες παραμετροποιημένες υπό μονάδες. Επίσης ορισμένες παράμετροι πάλι διαμέσου της Python ενημερώνονται

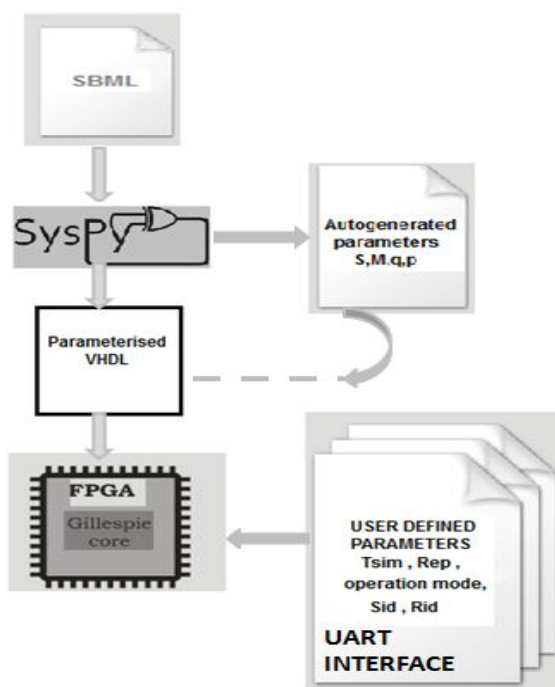
αυτόματα, όπως ο αριθμός των αντιδράσεων και το πλήθος των στοιχείων που ενημερώνονται απευθείας από το xml αρχείο που περιγράφει το biomodel. Ο αριθμός των αντιδράσεων M όπως αναλύσαμε έχει άμεση σχέση με τις μνήμες και συγκεκριμένα με το μέγεθος αρχικοποίησης όλων των μνημών (arrays). Επίσης ο αριθμός των στοιχείων έχει άμεση σχέση με το μέγεθος της μνήμης ST(species Table).

Το εάν θα επιλεγεί ο FRM ή ο NRM αλγόριθμος ή γενικότερα ο τρόπος λειτουργίας εξαρτάται από τις παραμέτρους που έχει περάσει ο χρήστης τα οποία αντανακλούν στον εσωτερικό προγραμματισμό της μονάδος. Ο προγραμματισμός γίνεται με τον καταχωρητή ελέγχου Control Register που φαίνεται παρακάτω



Σχήμα 48: Προγραμματισμός εσωτερικού Control register.

Πιο συγκεκριμένα το MSB bit (7) υποδηλώνει εάν είναι '1' να εκτελεστεί ο NRM και εάν '0' ο FRM. Ο τρόπος λειτουργίας ρυθμίζεται από τα 2 τελευταία bits το simindi και το mode και πιο συγκεκριμένα εάν το simindi = '1' τότε υλοποιείται ο 2^{ος} τρόπος λειτουργίας και εάν το simindi είναι '0' και το mode='1' τότε έχω τον 1^ο τρόπο λειτουργίας δηλαδή μια N παράλληλη προσομοίωση. Επίσης έχει προβλεφθεί η ύπαρξη εξαγωγής στατιστικών στοιχείων από την μονάδα STU όταν το bit (2) είναι '1' και μόλις το $t=t_{max}$ γίνεται η στατιστική επεξεργασία. Καθώς επίσης και τα bits (3) και (6) μπορούν να χρησιμοποιηθούν στην περίπτωση που θέλω προσομοίωση με υπό-τρόπους λειτουργίας δηλαδή στην περίπτωση που ο αριθμός των μοντέλων προς προσομοίωση είναι μικρότερος των PE. Τα bits 5 και 4 είναι δεσμευμένα από το σύστημα και χρησιμοποιούνται μόνο σε περιπτώσεις immediate reset. Επίσης υπήρχε και η πρόβλεψη άμεσης φόρτωσης δεδομένων στις μνήμες ROM μέσω σειριακής θύρας για την αλλαγή του περιεχομένου ώστε να αποφεύγεται η διαδικασία της πλήρους σύνθεσης εξ'αρχής. Ο πίνακας 10 παραθέτει τις παραμέτρους που χρειάζονται για να μεταβληθούν από τον χρήστη για τον προγραμματισμό της μονάδας καθώς επίσης και των μεταβλητών που ενημερώνονται αυτόματα από το SBML αρχείο μέσω της Python.



Σχήμα 49: Διαδικασία ανάθεσης παραμέτρων και σύνθεσης κυκλωμάτων ειδικού σκοπού μέσω της Python.

Πίνακας 10: Γενικοί Παράμετροι σχεδίασης.

Parameters	Operation
N	Number of Pes
M	Number of reactions
Tsim	Number of simulation runs
Rep	Number of Iterations (runs)
RNG seed	Random Number Generation seed
Algorithm	FRM or NRM (upcoming LDM,wNRM,ODM,wNRM)
Mode of operation (k)	1 st mode (one simulation in parallel) 2 nd mode(N different simulations in parallel) N/2 different in parallel simulations
Rid	Reaction index for statistics
Sid	Species Index for Statistics
Automatically generated parameters from the SBML model	
q	Number of reactants (Reaction order)
p	Number of products
M	Number of Reactions
m	Number of Species
Q	Maximum expected packets (estimated reaction cycles)

5.5 Στατιστικά στοιχεία για υλοποίηση σε Xilinx των SoCs FRM1X, FRM2X, FRM4X και FRM4X-NRM4X μετά την εισαγωγή του επεξεργαστή Leon (Users Interface)

Οι προηγούμενοι πίνακες παρείχαν στατιστικά στοιχεία για σχεδιάσεις FRMxX και FRM_NRM με x=1,2,4 και 8 PEs με αυτόνομα συστήματα επικοινωνίας με DDR2 και UART ελεγκτές όπως αναλύθηκε και στο κεφάλαιο 3. Παρακάτω παρατίθεται και ο πίνακας των στατιστικών στοιχείων με την εισαγωγή του επεξεργαστή Leon. Η ύπαρξη του Leon και όλων των υπό-μονάδων του (AMBA Bus και των controllers Ethernet, Uart, JTAG, DDR2,GPIO κλπ) επιβάρυνε την όλη σχεδίαση όσων αναφορά το διαθέσιμο χώρο στο

αναπτυξιακό. Για αυτό το λόγο μέσω του XST ρυθμίσαμε όλες τις παραμέτρους απλοποίησης για να μπορεί το σχέδιο να συνθεθεί για το Virtex 5 FPGA. Αυτό είχε σαν αποτέλεσμα να αυξηθεί ο χρόνος σύνθεσης σε περίπου 5.5 ώρες για την μεγάλη μονάδα FRM4X,NRM4X+Leon. Επίσης ο επεξεργαστής Leon έχει προδιαγραφές λειτουργίας μέχρι 80 MHz, όμως βρέθηκε ότι μέσω ειδικού κυκλώματος πολλαπλασιαστή συχνότητας μπορεί εσωτερικά να λειτουργήσει στα 160 MHz οπότε για την συγκεκριμένη λειτουργία και αφού τα συστήματα FRMxX με x=1,2,4,8 Pes και FRM4X-NRM4X λειτουργούν σαν σκλάβοι επάνω στο AMBA bus έπρεπε να κατέβει η συχνότητα λειτουργίας στα 160 Mhz. Με αποτέλεσμα κάποια ελάττωση της απόδοσης όπως φαίνεται στον πίνακα 11.

Πίνακας 11: Στατιστικά στοιχεία P.P.R προσομοίωσης για επιδόσεις Mreactions/sec, καθυστερήσεων και απόδοσης για την FRM1X-FRM2X-FRM4X-FRM4X_NRM4X SoC σχεδίαση σε Xilinx.

Core	FRM1X+Leon	FRM2X+Leon	FRM4X+Leon	FRM4XNRM4X+Leon
Reactions	512	1024	2048	2048
Frequency	160	160	160	160
Latency	Q (44+70)+9	Q (44+70)+11	Q (44+70)+13	Q (44+70)+15/log(N)/N*log(N)
Performance	145-139	263- 243	442-388	1011-780 770-625
LUTs	23394	35099	55642	60128
DSP	16	24	48	64
Regs	19077	33704	58082	64485
Ram-Blocks	59	61	97	116

ΚΕΦΑΛΑΙΟ 6

Αποτελέσματα και Στατιστικά Στοιχεία Προσομοίωσης

6.1 Επίδοση συστημάτων και Στατιστικά στοιχεία Προσομοίωσης

Σε αυτό το κεφάλαιο θα δούμε τις αποδόσεις και την επιτάχυνση λειτουργίας (speedup) για όλες τις FRM σχεδιάσεις και για όλες τις βασικές υπό μονάδες καθώς και για την σχεδίαση FRM/NRM σε σύγκριση με υλοποιήσεις σε CPUs της Intel και GPUs της INVIDIA από υλοποιήσεις των αλγορίθμων σε C++ για τις διάφορες CPU και σε CUDA για της GPUs. Επιπλέον θα αναδείξουμε την επιτάχυνση του αλγορίθμου του NRM σε Hardware σε σύγκριση πάντα με την μέχρι τώρα βιβλιογραφία.

6.2 Αποτελέσματα επιδόσεων για τα SoCs FRM1X-2X-4X και FRM4X+NRM4X σε σχέση με υλοποιήσεις του FRM αλγόριθμου σε CPUs της Intel

Στον πίνακα 12 φαίνονται τα speedups για τα FRM1X, FRM8X σε σύγκριση με υλοποίηση του FRM με κώδικα που έχει υλοποιηθεί σε C++ για τις CPU της Intel, Dual Core Duo 2, 4Gb Ram στα 2.16GHz και αντίστοιχα Octal Core της Intel i7 6 GB Ram στα 2.66 GHz στο DEV-C σε Windows για ένα μοντέλο 512 αντιδράσεων 2^{ου} βαθμού. Οι συγκρίσεις δίνουν ένα speedup για την απλή περίπτωση του FRM1X με ένα επεξεργαστή PE της τάξης του 80, ενώ με 8 επεξεργαστές και με σύγκριση για i7 CPU βρέθηκε speedup άνω των 300. Αυτό υποδηλώνει το πλεονέκτημα των Hardware υλοποιήσεων συγκρινόμενο με τις Software ειδικότερα για μεγάλα μοντέλα βιοχημικών αντιδράσεων.

Πίνακας 12: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση με FRM σε C++ για διάφορες CPU.

Model M =512 Reactions (2 nd order) Core	Reaction cycle time (μ-sec)	M Reactions /sec	Throughput (MRC/sec)
FRM1X	1.71	300	0.58
FRM8X	0.37	1365	2.66
Core 2	156.25	3.28	0.0064
i7	126.58	4.04	0.0079

Σε αυτό το κεφάλαιο θα δείξουμε την επιτάχυνση του NRM αλγορίθμου. Εάν παρατηρήσουμε τον αλγόριθμο στο κεφάλαιο 2 την εξίσωση 23 θα διαπιστώσουμε ότι το βήμα αυτό για την δικιά μας υλοποίησης είναι περιττό αφού θεωρούμαι απόλυτες χρονικές στιγμές και εφόσον σε 1 κύκλο αντίδρασης έχουμε μέσω του DG και του Data Path όλη την ενημέρωση για τις αντιδράσεις όπου τα στοιχεία τους θα μεταβληθούν στον κύκλο αντίδρασης. Άρα υπολογίζονται και όλες οι πιθανότητες αντίδρασης $a_j(x)$. Έτσι θα

υπολογισθούν μόνο όσοι χρόνοι αλλάξουν εγκαταλείποντας το βήμα 23. Επίσης μέσω της παράλληλης σχεδίασης του συστήματος πολλαπλών συγκρίσεων κατορθώσαμε και επιταχύνουμε περαιτέρω την διαδικασία. Ο πίνακας 13 δείχνει τους χρόνους για όλες τις διαδικασίες του NRM σε C++ για την προσομοίωση του μοντέλου 293 (316 αντιδράσεις 136 στοιχεία) και του HRS (Heat response of E-Coli 61 αντιδράσεις 28 στοιχεία), αντίστοιχα θα δούμε τις ίδιες διαδικασίες στο NRM4X SoC όπου καθαρά θα φανούν τα πλεονεκτήματα και η υπεροχή σε απόδοση σε επίπεδο υλικού. Ο συνολικός χρόνος είναι το άθροισμα όλων των χρόνων

Πίνακας 13: Χρόνοι υπολογισμού για όλες τις διαδικασίες του NRM σε C++ για την προσομοίωση των μοντέλων 293 και HRS σε επεξεργαστή i7 CPU της Intel.

Operation	NRM i7 (μ-sec) C++ 1 reaction cycle/ HRS	NRM i7 (μ-sec) C++ 1 reaction cycle/293
C_minimum	92.8/4.63*	169/23.9*
Cp+Cg	18.5	35
Crand	1.54	6.98
Ct	1.99	4.43
CupdateX	2.27	5.28
Ctotal	117.1/28.93	220/75.6

Ο συμβολισμός με (*) δείχνει αντίστοιχα 2 χρόνους για 2 διαφορετικούς τρόπους υπολογισμού C_minimum, ο 1^{ος} , είναι για την ενημέρωση του PQ και ο 2^{ος} για την αποθήκευση της πληροφορίας στις FIFO και από εκεί για την εύρεση του ελαχίστου. Ο πίνακας 14 δείχνει τα βήματα του NRM αλγορίθμου όπως προσομοιώθηκαν από το FRM4X_NRM4X SoC για τις προσομοιώσεις των μοντέλων 293 και HRS. Ο συνολικός χρόνος υπολογίζεται αθροίζοντας τους επιμέρους χρόνους των μαύρων σκιαγραφημένων τετραγώνων. Αντίστοιχα για τον χρόνο C_minimum παρατηρούμαι 3 διαφορετικούς χρόνους για τις 3 υλοποιήσεις που είναι ο 1^{ος} χρόνος για την απλή υλοποίηση με τις FIFO και τις βαθμίδες MTU1 και MTU2 και οι άλλοι 2 χρόνοι είναι αντίστοιχα για την πλήρη παράλληλη ταξινόμηση και για τα δένδρα ταξινόμησης. Στον Πίνακα 15 δίνονται οι χρόνοι για την διαδικασία προσομοίωσης με τον NRM αλγόριθμο για το μοντέλο 293 μέχρι ο χρόνος προσομοίωσης να φτάσει τα 90,000 δευτερόλεπτα σε σύγκριση με αντίστοιχη προσομοίωση σε CPU i7 Intel και GPU NVIDIA.

Πίνακας 14: Χρόνοι υπολογισμού για όλες τις διαδικασίες του NRM από το FRM_NRM4X SoC σε τρόπο λειτουργίας 1 για τον NRM αλγόριθμο για την προσομοίωση των μοντέλων 293 και HRS.

Operation/BIOMD	NRM4X (μ-sec) 1 reaction cycle /HRS	NRM4X (μ-sec) 1 reaction cycle 293
C_minimum	0.07/0.04/0.06	0. 3/0.054/0.56
Cp+Cmems	0. 082	(0.08- 0.22*)
Crand	0.015	0.034
Ct	0.17	(0.19-0.28*)
CupdateX	0.067	0.078
Ctotal	0.255/0.297/0.252	(0.56-0.32-0.83)/(0.66-0,4-0.92)

Ο πίνακας 16 δείχνει την σύγκριση ως προς την επίδοση μεταξύ των 2 αλγορίθμων FRM και NRM με 4 επεξεργαστές προσομοιώνοντας ένα μοντέλο 512 αντιδράσεων 2^{ου} βαθμού και 3^{ου} και μέγιστο αριθμό εξαρτήσεων 80 για το μοντέλο αυτό (80 αντιδράσεις μπορούν να επηρεαστούν από την πυροδότηση κάποιας εκ των M του δικτυώματος).

Πίνακας 15: Χρόνοι προσομοίωσης του μοντέλου 293 στο NRM4X SoC σε CPU και αντίστοιχα σε GPU.

Model	NRM C++(sec)	NRM4X SoC(sec)	GPU(sec)
BIOMD293	14.54	0.061	1.81

Πίνακας 16: Σύγκριση επιδόσεων μεταξύ των 2 προσομοιωτών NRM4X και FRM4X.

Component	FRM4X	NRM4X
Max Performance (MReactions/sec)	633*-720**	For D=80 1600* - 790**
Max Throughput (MReactions cycles/sec)	1.4*-1.23**	3.1* - 1.54**

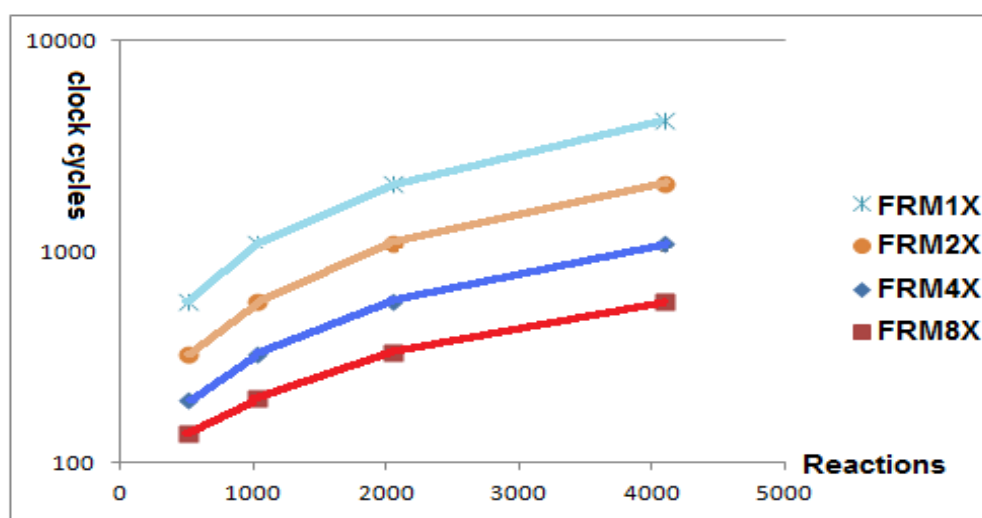
* 3^{ου} βαθμού αντίδραση με όλα τα στοιχεία ίδια για υλοποίηση στο Virtex 7.

**Μοντέλα 1^{ου} και 2^{ου} βαθμού για υλοποίηση σε Virtex.

Για την συγκεκριμένη μονάδα και από τον πίνακα 16 μπορούμε εύκολα να δούμε την υπεροχή του αλγορίθμου NRM. Ωστόσο η πολυπλοκότητα του συστήματος και ο καταμερισμός των στοιχείων στο FPGA σε μεγάλα δικτυώματα αντιδράσεων τείνει να το καταστήσουν απαγορευτικό. Επίσης όσο ο αριθμός D (αλληλεξαρτήσεις μεταξύ των αντιδράσεων) αυξάνει ο NRM τείνει να γίνει συγκρίσιμος με τον FRM.

Από τον σχήμα 50 μπορούμε να παρατηρήσουμε ότι για ένα σταθερό μοντέλο 512 βιοχημικών αντιδράσεων η απόδοση για τον αλγόριθμο FRM αυξάνει σχεδόν γραμμικά με την αύξηση των PEs δηλαδή με την αύξηση της παραλληλοποίησης του συστήματος. Το οποίο καθιστά τον FRM λόγο της εύκολης παραλληλοποίησης για τον 1^ο τρόπο λειτουργίας

τον πλέον κατάλληλο αλγόριθμο για πολύ μεγάλα δίκτυα βιοχημικών αντιδράσεων. Επίσης το ίδιο φαίνεται να ισχύει και στον NRM αλγόριθμο. Όμως όπως έχουμε αναφέρει και αναλύσει στο κεφάλαιο 4 είναι αρκετά δύσκολη η παραλληλοποίηση του καθώς απαιτεί πολύ υλικό (Hardware) και συνεπώς επιβάλλει μια πολύπλοκη υλοποίηση για την παραλληλοποίηση που σημαίνει ότι η σύνθεση τέτοιων κυκλωμάτων γίνεται αρκετά δύσκολη για αριθμό PEs άνω του 4. Το παρακάτω διάγραμμα δίνει την γραμμική εξάρτηση της απόδοσης με βάση των χρόνων κύκλων λειτουργίας για πολλά PEs για την περίπτωση του FRM αλγορίθμου.



Σχήμα 50 : Διάγραμμα απόδοσης εξαρτώμενο από τον αριθμό των PEs για μεγάλα βιοχημικά μοντέλα αντιδράσεων.

Από το διάγραμμα του σχήματος 50 παρατηρούμαι ότι καθώς αυξάνεται ο αριθμός των PEs μειώνεται το Latency σε clock cycles και αυξάνεται η απόδοση για συστήματα πολλαπλών αντιδράσεων. Πιο συγκεκριμένα για ένα μοντέλο 512 αντιδράσεων 2^{ου} βαθμού η απόδοση είναι 2.6 η οποία με βάση τη διεθνής βιβλιογραφία αποδεικνύεται ως η μεγαλύτερη απόδοση σε σχεδιάσεις τέτοιου τύπου. Το ίδιο ισχύει και για την σχεδίαση με τον NRM όπου εμφανίζεται μέγιστη απόδοση 3.1 η οποία επίσης αναδεικνύεται ως η μέγιστη παρατηρούμενη με βάση και την διεθνής βιβλιογραφία [5] [15].

ΚΕΦΑΛΑΙΟ 7

Έλεγχος ορθότητας λειτουργίας

Στο παρών κεφάλαιο θα αναλύσουμε τον τρόπο με τον οποίο έγινε ο έλεγχος (testing) για να διαπιστωθεί η σωστή λειτουργία των συστημάτων FRMxX /NRMxX και SysPy σε πραγματικό χρόνο. Με βάση την ανάλυση που κάναμε σε προηγούμενα κεφάλαια για τον προγραμματισμό της πλατφόρμας το σύστημα SysPy αναλαμβάνει το κομμάτι της επικοινωνίας με το Η/Υ. Το SysPy καλεί συναρτήσεις σε C οι οποίες προγραμματίζουν τον επεξεργαστή Leon. Αυτό γίνεται αφού έχει γίνει σύνθεση για το σύστημα FRMxX+Leon ή FRM4X_NRM4X+Leon με βάση το εκάστοτε SBML μοντέλο (δηλαδή αρχικοποίηση των μνημών) και τον προγραμματισμό της πλατφόρμας με βάση τις γενικές παραμέτρους (generic Parameters) που έχουν διαμορφωθεί με βάση πάντα τις ανάγκες του χρήστη. Έπειτα μέσω GRMON και προγράμματος σε Python προγραμματίζεται μέσω του FPGA ο επεξεργαστής Leon με πρόγραμμα γραμμένο σε C που καλείται μέσω της Python. Έπειτα μέσω του SysPy γίνεται η μεταφορά των δεδομένων μέσω της θύρας Ethernet από το FPGA στον Η/Υ. Τα δεδομένα στέλνονται σε μορφή πακέτων γράφονται στο δίσκο του Η/Υ και εμφανίζονται σε ένα αρχείο Matlab από το οποίο μέσω ειδικού προγράμματος δημιουργείται ένας πίνακας τιμών για την κάθε επανάληψη της προσομοίωσης από το οποίο τελικά δημιουργούνται οι κυματομορφές που δείχνουν τη συμπεριφορά όλων των στοιχείων της αντίδρασης.

7.1 Τρόπος προγραμματισμού και ελέγχου του συστήματος

Πιο αναλυτικά ο χρήστης με βάση τον Πίνακα 10 και αφού κατεβάσει από την Biomodules Database και διαλέξει το SBML αρχείο που θέλει να προσομοιωθεί περνάει μέσω προγράμματος επικοινωνίας (GUI interface) τις απαραίτητες παραμέτρους οι οποίες μεταφράζονται σε μεταβλητές που εισέρχονται στον κώδικα VHDL απευθείας. Μέσα από αυτές τις παραμέτρους επιλέγεται και ποια σχεδίαση θα συνθεθεί στον XST δηλαδή εάν θα είναι η FRM1X,2X,4X-8X ή FRM_NRM4X το οποίο γίνεται με συνδυασμό των παραμέτρων επιλογής του αλγορίθμου (Algorithm FRM/NRM) και της μεταβλητής P που δείχνει τον αριθμό των επεξεργαστών PEs. Το SysPy έπειτα δημιουργεί την σύνδεση του Leon και των FRMxX ή FRM_NRM4X SoC σε VHDL όπου δημιουργείται ένας φάκελος που περιέχει όλο το αρχείο προς σύνθεση. Αυτό γίνεται μέσω TCL script τα οποία καλούν τις κατάλληλες υπομονάδες (VHDL κώδικας) μέσα από μια βιβλιοθήκη που έχει φτιαχτεί (με την κατάλληλη αρχικοποίηση των μνημών για το κάθε SBML μοντέλο). Επίσης καλείται και ο

γενικός κώδικας που ενώνει το top Entity με τον επεξεργαστή Leon σε VHDL. Με αυτόν τον τρόπο συνδέονται όλα τα κομμάτια αυτόματα και μέσω ειδικών εντολών που χρησιμοποιούνται από την Xilinx γίνεται η δημιουργία αρχείου έτοιμου προς σύνθεση. Έπειτα πάλι με εντολές τύπου make και run αρχίζει η σύνθεση για την δημιουργία .bit αρχείου αυτόματα για τον προγραμματισμό του FPGA. Έτσι ο χρήστης το μόνο που πρέπει να κάνει είναι να διαλέξει το μοντέλο, να περάσει τις παραμέτρους και να διαλέξει τον αριθμό των πακέτων που θέλει να δει. Το τελευταίο βήμα ουσιαστικά δηλώνει και το πόσες επαναλήψεις του αλγορίθμου θέλει να δει. Έτσι για ένα μοντέλο ενζυματικών αντιδράσεων μπορεί να ζητήσει 4,000 πακέτα τα οποία ισοδυναμούν με 8 προσομοιώσεις του αλγορίθμου. Έπειτα ο χρήστης θα πρέπει να τρέξει τους αναπτυσθέντες κώδικες σε Matlab και Python για να βγουν οι κυματομορφές.

7.2 Biomodels που χρησιμοποιήθηκαν για το testing και η συνοπτική ανάλυση τους

Για τον έλεγχο του συστήματος χρησιμοποιήθηκαν διάφορα Βιολογικά μοντέλα όπως απλά μοντέλα που με βάση την βιβλιογραφία κατασκευάσαμε σαν το Lotka_Volterra [16] και αντιδράσεις με ένζυμα [17], τα οποία μπορούν γρήγορα να δώσουν μια εικόνα για το εάν ο προσομοιωτής λειτουργεί σωστά. Έπειτα χρησιμοποιήθηκαν μοντέλα από την Biomodels database όπως τα BIOMD000000188-189 [18] το Heat Response Shock of E-Coli [19], κάποια μοντέλα που περιγράφουν αυτοελεγχόμενους γονιδιακούς μετασχηματισμούς σε πρωτεΐνες [30] μοντέλα με πολλές αντιδράσεις για την προσομοίωση του ρόλου της πρωτεΐνης Asyn στην ασθένεια του Parkinson [21] καθώς επίσης και τη διαδικασία παραγωγής του ενζύμου UCHL1[22] στην δημιουργία και εκφύλιση της νόσου του Πάρκινσον.

Για τον έλεγχο των αποτελεσμάτων πάρθηκαν από το κάθε μοντέλο πολλά δεδομένα από πολλαπλές προσομοιώσεις για το κάθε στοιχείο σε έναν πίνακα δεδομένων σε Matlab που δείχνει την εξέλιξη όλων των στοιχείων για το κάθε στιγμιότυπο χρόνου και από εκεί βγήκαν οι μέσες τιμές για τα στοιχεία ενδιαφέροντος, όπου μέσω Matlab έγινε το plotting για το κάθε μοντέλο. Οι κυματομορφές αυτές που δημιουργήθηκαν στην Matlab συγκρίθηκαν με τα ίδια δεδομένα που παρήγαγαν οι προσομοιωτές Copasi και Ibiosim tool καθώς επίσης και με στοιχεία από την διεθνή βιβλιογραφία για τα συγκεκριμένα μοντέλα για να αποδειχθεί η φερεγγυότητα των αποτελεσμάτων, όπως θα δούμε από τις αναλύσεις που παρουσιάζομαι στην επόμενη παράγραφο. Επίσης σε σχέση με τους προσομοιωτές Copasi και Ibiosim έγινε εκτίμηση του χρόνου speedup του συστήματος μας σε σχέση με τα εργαλεία προσομοίωσης. Μετρήθηκαν δύο χρόνοι, ο ένας καθαρά για το σύστημα

FRMxX +Leon, για την εγγραφή των δεδομένων στην DDR2 του FPGA μέσω του Leon και ο άλλος για μια end to end διαδικασία η οποία περιλαμβάνει και το χρόνο μεταφοράς από την DDR2 μνήμη στην Ethernet του FPGA και από την Ethernet στο PC καθώς προσμετρείται και ο χρόνος εγγραφής στο δίσκο στο τερματικό για την σχεδίαση της κυματομορφής μέσω Matlab.

Αξίζει να σημειωθεί ότι δεν χρησιμοποιήθηκε για την μεταφορά η θύρα Ethernet του 1GB/sec που δίδεται έτοιμη για τον επεξεργαστή Leon αλλά η open- core έκδοση των 100Mbit/sec και αυτό πρέπει να το τονίσουμε διότι θα είχαμε μεγαλύτερη αύξηση του speedup όπως είναι φυσικό χρησιμοποιώντας την 1GB/sec θύρα Ethernet.

Επιπλέον για όλα τα μοντέλα έχει γίνει εκτίμηση του τετραγώνου του μέσου συστηματικού σφάλματος MSE τόσο για τον αλγόριθμο FRM αλλά και για τον NRM σε σχέση με τις αντίστοιχες παρομοιώσεις στον προσομοιωτή Corasi για τους ίδιους χρόνους, συνθήκες προσομοίωσης καθώς και για τον ίδιο αριθμό επαναλήψεων.

7.3 Συγκρίσεις προσομοιώσεων των SOC's με MATLAB, Copasi, Ibiosim tool

Πριν αρχίσουμε τις αναλύσεις των πειραματικών στοιχείων από τις διάφορες προσομοιώσεις θα επικεντρωθούμε στον τρόπο με τον οποίο έγιναν οι συγκρίσεις για τον έλεγχο της ορθότητας του προσομοιωτή. Για την ανάκτηση και την ανάλυση των αποτελεσμάτων δημιουργήθηκε ένα πρόγραμμα σε Matlab το οποίο από του πίνακες που έχουν δημιουργηθεί μέσα από το SysPy για όλους του κύκλους αντιδράσεων που αναλύθηκαν συνθέτει την κατάσταση του συστήματος για όλα τα στιγμιότυπα χρόνου ξέροντας τις συγκεντρώσεις όλων των ενδιαφερόμενων στοιχείων προς προσομοίωση.

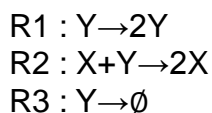
Έχοντας το προηγούμενο ως δεδομένο συλλέγουμε για τα ίδια αρχικά ανύσματα στις γεννήτριες παραγωγής τυχαίων αριθμών του Copasi και των FRM_NRM SoCs τις συγκεντρώσεις των στοιχείων την κάθε χρονική στιγμή για όλες τις προσομοιώσεις.

Η μέθοδος προσομοίωσης με το εργαλείο Copasi κάνει αναγωγή και δειγματοληψία των αποτελεσμάτων στο χρόνο (sub-sampling) για να μειώνονται οι απαιτήσεις σε μνήμη καθώς και για μείωση του χρόνου της διαδικασίας. Αυτό πρακτικά σημαίνει ότι ο χρήστης ορίζει χρονικά διαστήματα (βήμα χρόνου ανά 50) για την εξαγωγή των συγκεντρώσεων. Το σύστημα μας εξάγει από τη θύρα Ethernet τα δεδομένα σε κάθε χρονική στιγμή χωρίς δειγματοληψία. Για να γίνει λοιπόν σύγκριση των αποτελεσμάτων δημιουργήσαμε ένα MATLAB κώδικα ο οποίος με βάση του πίνακες των δεδομένων από το Copasi κάνει αναγωγή στο χρόνο των αποτελεσμάτων από την πλατφόρμα προσομοίωσης στο FPGA. Έτσι εάν έχουμε ορίσει μία βάση χρόνου στο Copasi ανά 50 sec θα πρέπει από όλους του κύκλους αντίδρασης από τις προσομοιώσεις στα SoC στο FPGA να πάρουμε τις συγκεντρώσεις των στοιχείων στις αντίστοιχες χρονικές στιγμές που προσεγγίζουν τις χρονικές στιγμές από το Copasi.

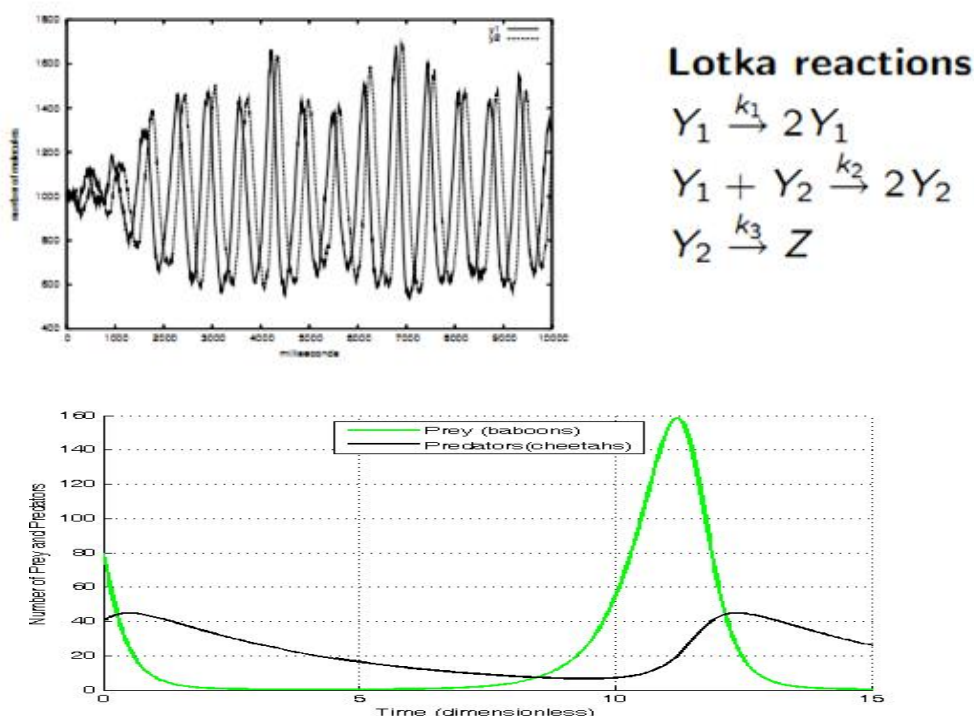
Με βάση την προηγούμενη ανάλυση για την εύρεση του τετραγώνου του μέσου συστηματικού σφάλματος (MSE) παίρνουμε τις μέσες τιμές των συγκεντρώσεων από όλες τις προσομοιώσεις που έχουν γίνει στο Copasi και τις αφαιρούμε από τις αντίστοιχες μέσες τιμές στις αντίστοιχες χρονικές στιγμές από τους προσομοιωτές των SSA SoCs και τις υψώνουμε εις το τετράγωνο για τον υπολογισμό του MSE.

$$MSE = (X'_i - X''_i)^2 \quad (29)$$

Στο κεφάλαιο αυτό θα αρχίσουμε την ανάλυση μας με το 1^ο απλό ευρέος διαδεδομένο βιοχημικό μοντέλο το Lotka_Volterra. Όλες οι προσομοιώσεις που θα δούμε δείχνουν μέσες τιμές συγκεντρώσεων. Το μοντέλο Lotka αποτελείται από 3 αντιδράσεις των 2 στοιχείων όπως φαίνεται και παρακάτω.

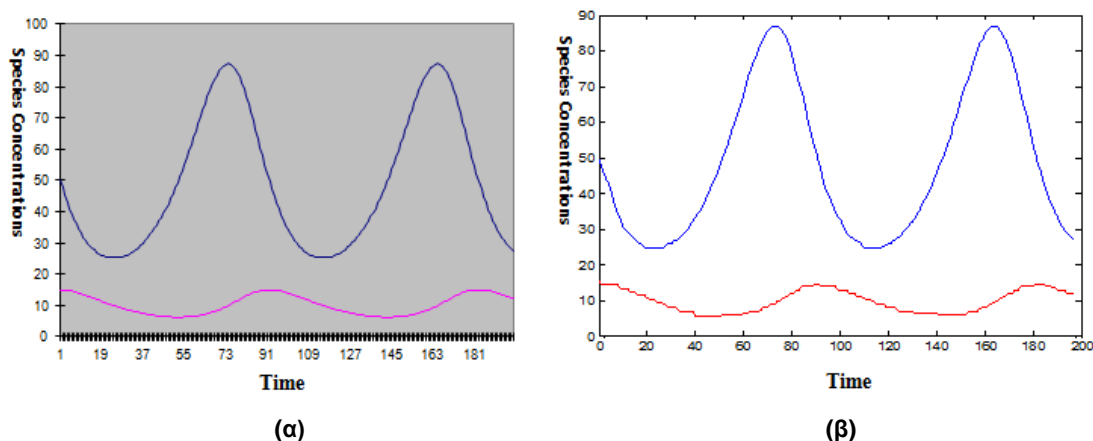


Μοντέλα αυτής της μορφής εκτελούν ταλαντώσεις γύρω από καθορισμένες τιμές και χαρακτηρίζονται ως συστήματα τα οποία έχουν ανάδραση. Το μοντέλο αυτό είναι αρκετά γνωστό από μαθηματικής πλευράς καθώς έχουν γίνει πολλές αναλύσεις μαθηματικές οι οποίες προσπαθούν να περιγράψουν το φαινόμενο των ταλαντώσεων. Στο μοντέλο Lotka έχουμε 2 στοιχεία εκ των οποίων το ένα παρασιτεί και αυξάνει εις βάρος του άλλου, όμως υπάρχει ανάδραση στο σύστημα μας γιατί όταν αντιδρούν τα 2 στοιχεία δηλαδή όταν καταναλώνεται το ένα για να δημιουργηθεί το άλλο πάντα ένα μέρος του αρχικού παραμένει όπως φαίνεται από την αντίδραση R2 και αυτό έχει ως συνέπεια τη δημιουργία συνεχών ταλαντώσεων. Τέτοια συστήματα αυτοταλαντευόμενων συστημάτων υπάρχουν παντού στη φύση και οι βιολόγοι και οι μαθηματικοί χρησιμοποιούν την ανάλυση αυτού του μοντέλου για να τα περιγράψουν. Η εξέλιξη του φαινομένου εξαρτάται από τα στοιχεία των συγκεντρώσεων των βιοχημικών αντιδράσεων και του αριθμού k (rate της αντίδρασης). Παραδείγματα τέτοιων μοντέλων από την διεθνή βιβλιογραφία φαίνονται Στο σχήμα 51.



Σχήμα 51: Ταλαντώσεις Lotka Volterra για μελέτες ταλαντευόμενου συστημάτων με 2 στοιχεία. Η 1^η εικόνα περιγράφει ένα ταλαντευόμενο σύστημα για $X(0)=Y(0)=1000$. Το κάτω σύστημα είναι για μελέτη πληθυσμών 2 είδη πιθήκων που οι πληθυσμοί τους παρουσιάζουν ταλαντώσεις για $X(0)=40$ και $Y(0)=80$.

Έγιναν αναλύσεις 2 συστημάτων Lotka. Η 1^η ανάλυση αναφέρετε σε ένα μοντέλο 3 αντιδράσεων 2 στοιχείων το οποίο πάρθηκε από την βιβλιογραφία. Το μοντέλο αυτό ελέγχθηκε στο FRM4X SoC στο δεύτερο τρόπο λειτουργίας όπου ο χρόνος προσομοίωσης ανήλθε στα 181 δευτερόλεπτα και τα αποτελέσματα φαίνονται στο σχήμα 52 για 10 επαναλήψεις του FRM. Το αποτέλεσμα της προσομοίωσης από την βιβλιογραφία αναλύει το μοντέλο με την Direct Μέθοδο του Gillespie στα 181 δευτερόλεπτα.



Σχήμα 52: (α) Ταλαντώσεις Lotka-Voltera από βιβλιογραφία, (β) προσομοίωση μοντέλου από πλατφόρμα FRM4X με επεξεργαστή Leon σε τρόπο λειτουργίας '2' για 10 επαναλήψεις και χρόνο προσομοίωσης 181 δευτερόλεπτα.

Ο πίνακας 17 παρουσιάζει τις επιδόσεις του συστήματος FRM4XSoC και Leon μέχρι τα αποτελέσματα να εγγραφούν στην μνήμη DDR2 για το SoC καθώς και για να μεταφερθούν μέσω θύρας Ethernet στο τερματικό.

Πίνακας 17: Απόδοση συστήματος FRM4X SoC+Leon για το Lotka-Voltera μοντέλο για μία end to end διαδικασία.

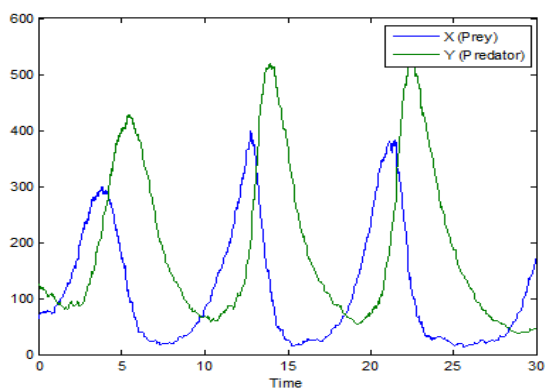
Model: Lotka Voltera	SoC Performance			End to End performance		
Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec	MRC/sec
452500	6	0,226	0,075	73.7	0,0078	0,0026

Το 2^ο μοντέλο που αναλύθηκε έχει περιγραφεί από το Πανεπιστήμιο της Πενσυλβάνιας και έχει γίνει προσομοίωση σε Matlab με κώδικα για την Direct Method όπως φαίνεται από το σχήμα 53(α).

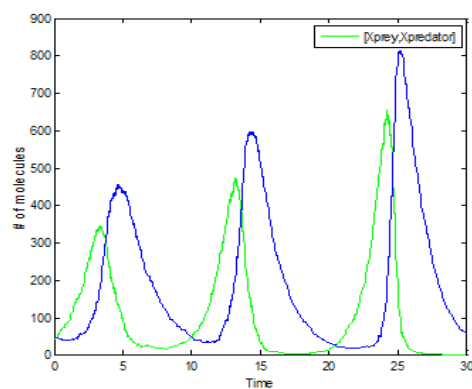
Στα παρακάτω διαγράμματα φαίνονται τα αποτελέσματα από τις προσομοιώσεις που έγιναν με τα FRM4X SoC +Leon για το σύστημα επικοινωνίας μέσω του SysPy όπως περιγράφηκε και αναλύθηκε σε προηγούμενη παράγραφο. Αξίζει να σημειωθεί ότι το συγκεκριμένο μοντέλο προσομοιώθηκε σε όλα τα FRMxX συστήματα καθώς και στο σύστημα FRM4X+NRM4X και οι συγκρίσεις των προσομοιώσεων έδειξαν σωστή λειτουργία σαν και αυτή που φαίνεται στα παρακάτω διάγραμμα (σχήμα 53).

Πίνακας 18: Απόδοση συστήματος FRM4X SoC+Leon για το Lotka-Voltera2 μοντέλο για μία end to end διαδικασία.

Model: Lotka Voltera	SoC Performance			End to End performance		
Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec	MRC/sec
125000	1.75	0,214	0,071	48	0,0078	0,026



(α)



(β)

Σχήμα 53: (α) Ταλαντώσεις Lotka Voltera από την προσομοίωση σε Matlab για τον αλγόριθμο του Gillespie με την Direct Method SSA όπου παρατηρούμαι ταλαντώσεις για αρχικές συγκεντρώσεις $X(0)=50$, $Y(0)=100$ και (β) ταλαντώσεις του ίδιου μοντέλου από το FRM4X SoC. Έγιναν και στις 2 προσομοιώσεις 10 FRM επαναλήψεις για χρόνο 30 δευτερόλεπτων.

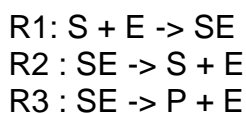
Παρατηρούμαι στο σχήμα 53 (β) την προσομοίωση από το FRM4X SoC σε τρόπο λειτουργίας '2' για 10 επαναλήψεις. Όλες οι γραφικές παραστάσεις αφορούν τις μέσες τιμές των συγκεντρώσεων από όλες τις προσομοιώσεις. Ο πίνακας 18 ένα αντιστοιχία με τον πίνακα 17 δείχνει τις επιδόσεις τους συστήματος για αυτό το μοντέλο.

Τα παρακάτω διαγράμματα έγιναν για το FRM4X για χρόνο αντίδρασης $T_{sim} = 50$ sec και για τον 2^ο τρόπο λειτουργίας με (*Algorithm k='1'*). Ο αριθμός των προσομοιώσεων είναι 10 για ένα πακέτο των 125,000 το οποίο πρέπει να ζητήσει ο χρήστης για να δει τις 10 προσομοιώσεις. Τα 125,000 πάρθηκαν σε χρόνο 1.75 sec στην DDR2 του Leon άρα η απόδοση είναι 214,285 Reactions/sec που σημαίνει ότι σε 1 sec έγιναν 214,285 αντιδράσεις. Για να καταγραφούν τα δεδομένα στο δίσκο θέλουμε 2,600 πακέτα το δευτερόλεπτο το οποίο σημαίνει ότι χρειάζονται 48 δευτερόλεπτα για τις 10 προσομοιώσεις και άρα έχουμε μία απόδοση της τάξης των 7,812 αντιδράσεων το δευτερόλεπτο.

Παρατηρούμαι ταύτιση συμπεριφοράς μεταξύ των προσομοιώσεων του FRM4X SoC σε σχέση με την αντίστοιχη προσομοίωση σε Matlab για το ίδιο μοντέλο με τις ίδιες αρχικές συνθήκες και παραμέτρους (σύγκριση εικόνων 52 και 53).

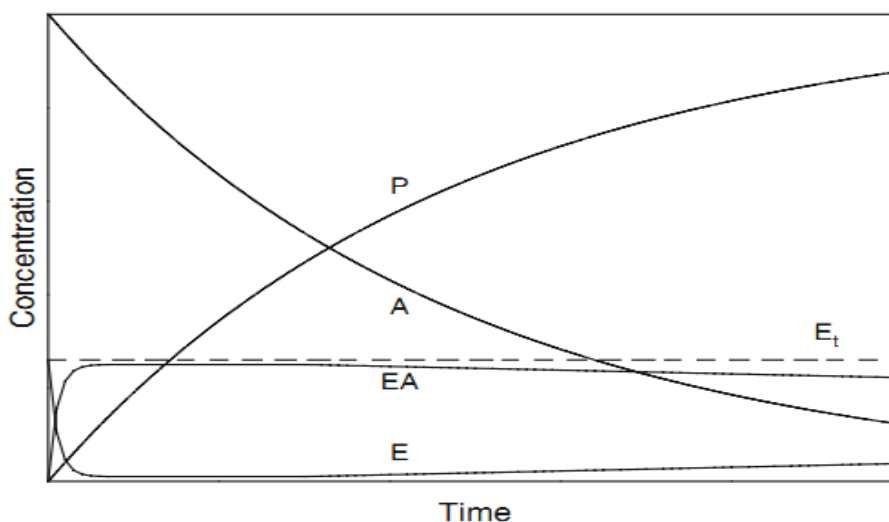
7.4 Περιγραφή μοντέλου ενζυματικών αντιδράσεων

Το επόμενο μοντέλο προς ανάλυση είναι το σύστημα ενζυματικών αντιδράσεων [12] επίσης πολύ χαρακτηρισμό στην Βιοχημεία για την περιγραφή μοντέλων συμπεριφοράς της κινητικής θεωρίας ενζύμων. Αυτό το σύστημα αποτελείται από 3 αντιδράσεις των 4 στοιχείων όπως φαίνεται παρακάτω



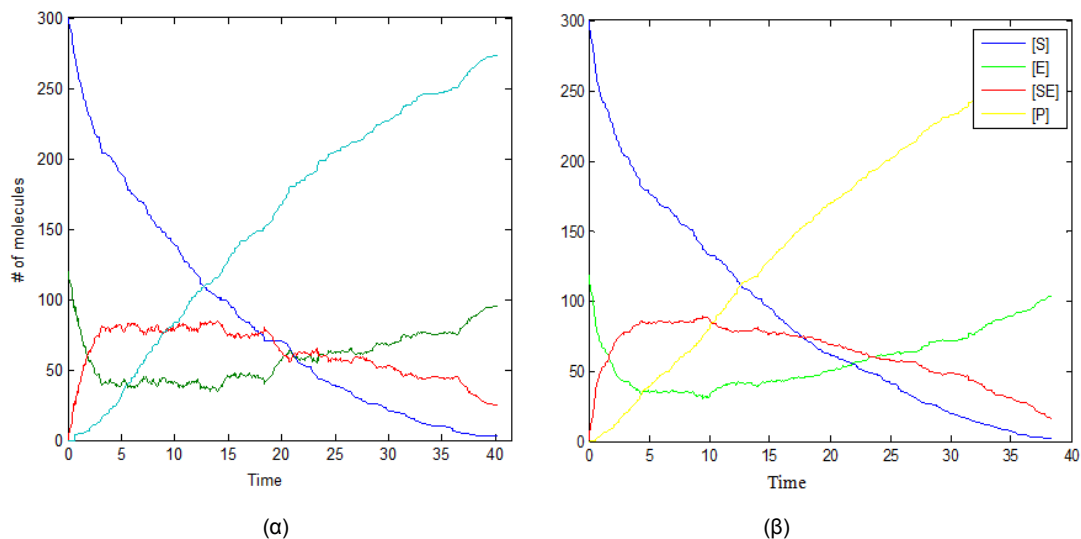
Τα ένζυμα όπως γνωρίζουμε είναι στοιχεία τα οποία βοηθούν και επιταχύνουν βιοχημικές αντιδράσεις. Με βάση λοιπόν τα ένζυμα τα αντιδρώντα αντιδρούν για την παραγωγή προϊόντων. Η πιο απλή ενζυματική αντίδραση είναι αυτή που προσομοιώσαμε και έχει να κάνει με την μετατροπή μιας χαρακτηριστικής ουσίας S σε μια ουσία P. Το ένζυμο θα αντιδράσει με το P θα παράγει ένα ενδιάμεσο προϊόν SE το οποίο θα αυτοκαταληθεί για να παράγει το P.

Από την διεθνής βιβλιογραφία παραθέτουμε την συμπεριφορά της λειτουργίας για μια τέτοιου είδους τύπου ενζυματική αντίδρασης όπως φαίνεται στο παρακάτω διάγραμμα (σχήμα 54).



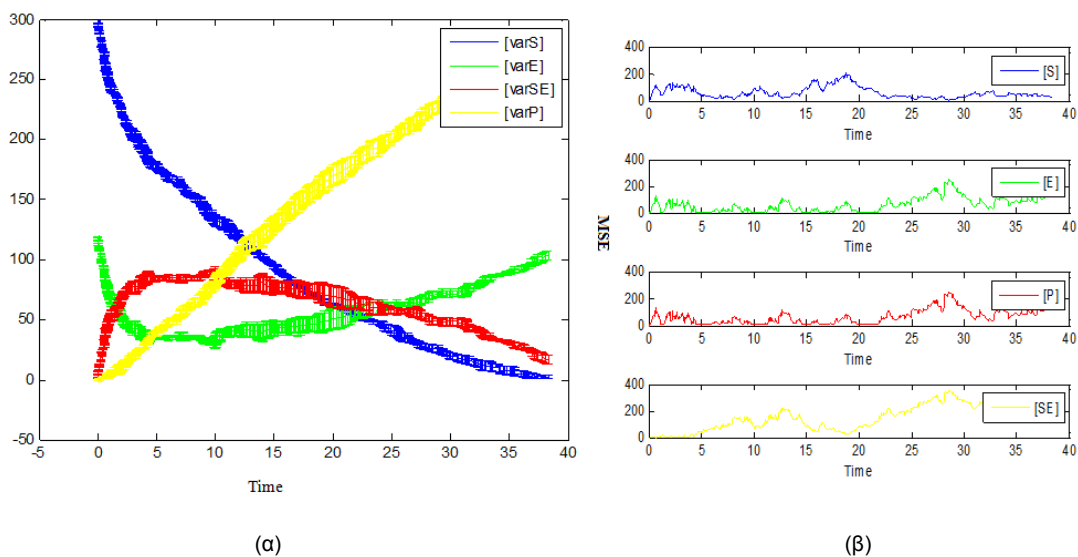
Σχήμα 54: Συμπεριφορά τυπικής ενζυματικής αντίδρασης.

Αυτό το μοντέλο ελέγχθηκε και προσομοιώθηκε με το FRM4X σε τρόπο λειτουργίας 2 δηλαδή για 4 ανεξάρτητες προσομοιώσεις. Ο χρόνος προσομοίωσης παράμετρος T_{sim} ορίστηκε στα 40 sec για 20 επαναλήψεις του αλγορίθμου όπου χρειάστηκαν 4,000 πακέτα. Τα αποτελέσματα της προσομοίωσης από το SoC φαίνονται στο σχήμα 55.



Σχήμα 55: (α) Συμπεριφορά ενζυματικών αντιδράσεων αποτελέσματα από την προσομοίωση σε Matlab (Direct Method) για χρόνο αντίδρασης 40 sec , 10 επαναλήψεις , (β) αποτελέσματα προσομοίωσης από το FRM4X SoC τρόπο λειτουργίας 2 χρόνο προσομοίωσης 40 sec 10 επαναλήψεις.

Η παρακάτω κυματομορφές δείχνουν σε κάθε χρονική το μέσο τετραγωνικό σφάλμα (MSE) στη σύγκριση με τον προσομοιωτή Corasi για όλο το φάσμα αντίδρασης καθώς και την διασπορά των τιμών από όλες τις επαναλήψεις.



Σχήμα 56: (α) Διασπορά των τιμών των συγκεντρώσεων από τις 10 επαναλήψεις (β) Μέσο τετραγωνικό σφάλμα για όλο το φάσμα αντίδρασης.

Παρατηρούμαι ότι και στο 2^ο πείραμα τα αποτελέσματα από το Matlab με το FRM4X συμπίπτουν, το οποίο είναι μια δεύτερη ένδειξη της ορθής λειτουργίας του προσομοιωτή. Για την ανάλυση αυτή χρειάστηκαν 4,000 πακέτα σε 0.056 sec για την καταγραφή των δεδομένων στην DDR2 μνήμη του Leon με απόδοση της τάξης των 214,285 Reactions/sec ενώ προσμετρώντας και τον χρόνο εγγραφής από την θύρα Ethernet στον δίσκο πάρθηκαν 4,000 πακέτα σε 1.53 δευτερόλεπτα και εγγράφηκαν στο δίσκο με συνολική απόδοση 18,300 Reactions/sec.

Πίνακας 19: Απόδοση συστήματος FRM4X SoC+Leon για την Ενζυματική αντίδραση για μία end to end διαδικασία.

Model: Lotka Voltera	SoC Performance			End to End performance		
	Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec
4000	0.056	0,214	0,0713	1.53	0,018	0,006

Παρατηρούμαι και πάλι ταύτιση συμπεριφοράς μεταξύ των προσομοιώσεων του FRM4X SoC σε σχέση με την αντίστοιχη προσομοίωση σε Matlab για το ίδιο μοντέλο με τις ίδιες αρχικές συνθήκες και παραμέτρους (σύγκριση σχημάτων 55 (α), (β)). Επίσης στον πίνακα 20 δείχνουμε το μέσο σφάλμα και το μέσο τετραγωνικό σφάλμα MSE για την κάθε χρονική στιγμή.

Πίνακας 20: Μέσο τετραγωνικό σφάλμα και μέσο σφάλμα για τα στοιχεία ενδιαφέροντος.

	MSE	Mean Error
S	67.04	8.1
E	53.4	7.3
P	53.4	7.3
SE	117.5	10.1

7.5 Προσομοίωση Biomodel 188 και 189

Συνεχίζοντας με τα πειράματά μας αναλύσαμε βιολογικά μοντέλα από την Biomodels Database. Τα συγκεκριμένα μοντέλα είναι τα Biomodels 188-189. Αυτά τα μοντέλα περιγράφουν την συμπεριφορά των συγκεντρώσεων των πρωτεϊνών p53 και Mdm2 οι οποίες έχουν αποδειχθεί ότι σχετίζονται με την δημιουργία καρκινικών όγκων.

Στα ανθρώπινα κύτταρα έχει παρουσιαστεί η δημιουργία της πρωτεΐνης p53 μετά από κάποιο λάθος στην αλληλουχία των βάσεων στο DNA. Η Mdm2 πρωτεΐνη η οποία ελέγχει την υπέρ-έκθεση της p53 αντιδρώντας με αυτή, δημιουργεί ένα ταλαντευόμενο σύστημα p53 και πρωτεΐνης Mdm2. Στο μοντέλο αυτό περιγράφεται με βάση την στοχαστική προσομοίωση η διαδικασία των ταλαντώσεων. Δηλαδή όταν εκτελείται μία βλάβη στην αλληλουχία DNA όπου αρχίζει και παρατηρείται υπέρ-έκθεση της πρωτεΐνης p53. Η πρωτεΐνη p14ARF δημιουργεί την Mdm2 η οποία ελέγχει την πρωτεΐνη p53. Η ύπαρξη της p53 έχει συνδεθεί με την ύπαρξη όγκων στο κύτταρο. Περιγράφονται λοιπόν δύο μοντέλα τα οποία έχουν σχέση με τον τρόπο έναυσης των ταλαντώσεων για τις πρωτεΐνες p53 και Mdm2. Ουσιαστικά υπάρχουν 2 διαφορετικοί μηχανισμοί ταλάντωσης ο ARF και ο ATM για τις ταλαντώσεις των πρωτεϊνών που όμως διαφέρουν ως προς τα στοιχεία που αντιδρούν. Τα μοντέλα 188-189 αποτελούνται αντίστοιχα από 20 αντιδράσεις των 20 στοιχείων και 19

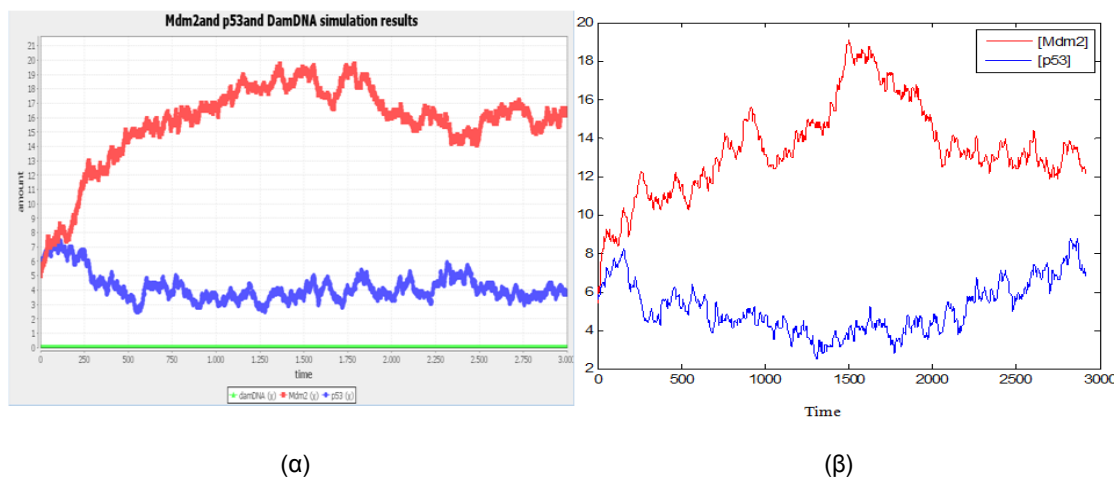
αντιδράσεις των 18 στοιχείων.

Τα παρακάτω διαγράμματα αφορούν την προσομοίωση των Biomodel 188 και 189 με βάση την προσομοίωση που έγινε στο Ibiosim tool για 8 επαναλήψεις του Gillespie DM αλγόριθμου όπου για τα μοντέλα 188-189 θέσαμε το (χρόνο προσομοίωσης αντίδρασης) αντίστοιχα παράμετρος Tsim στα 3000 sec και Tsim στα 2500 sec.

Το παρακάτω διάγραμμα δείχνει την προσομοίωση του βιοχημικού μοντέλου (20 αντιδράσεις των 20 στοιχείων) το οποίο έγινε στο FRM4X για χρόνο προσομοίωσης Tsim =3000 sec σε τρόπο λειτουργίας '1' (παράμετρος k='0') δηλαδή για 8 προσομοιώσεις του αλγορίθμου για ένα πακέτο των 4,000 δεδομένων.

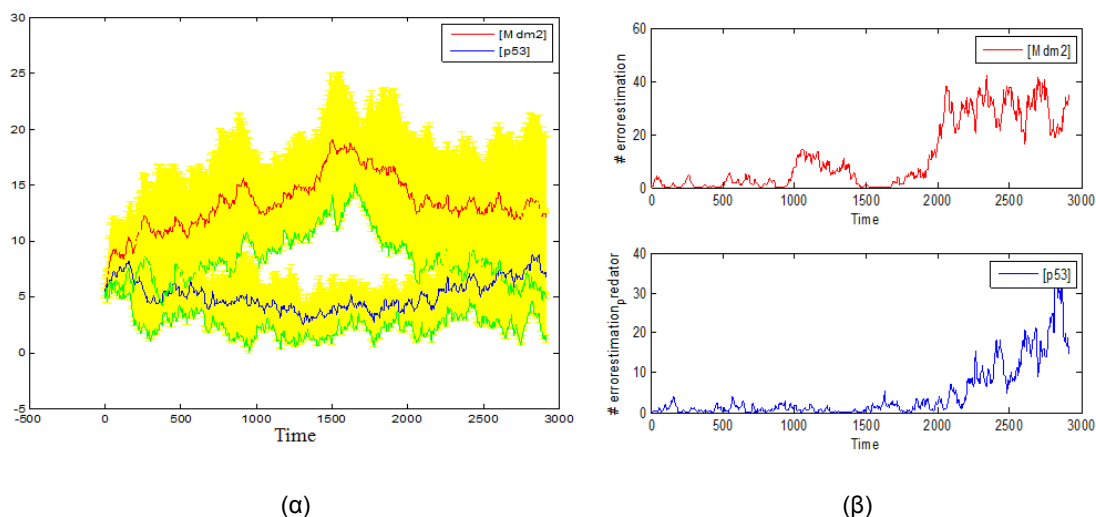
Οι συγκρίσεις έγιναν για τον ίδιο χρόνο Tsim και για τον ίδιο αριθμό επαναλήψεων από τον προσομοιωτή Ibiosim. Παρακάτω φαίνονται οι προσομοιώσεις για τις πρωτεΐνες Mdm2 , p53 και άλλων στοιχείων.

Το σχήμα 57 (α) δείχνει την προσομοίωση του μοντέλου 188 από το εργαλείο προσομοίωσης του Ibiosim tool για 8 επαναλήψεις του FRM αλγορίθμου. Αντίστοιχα το σχήμα 57(β) δείχνει την προσομοίωση από την βαθμίδα FRM4X SoC σε τρόπο λειτουργίας '1' (1 παράλληλη προσομοίωση) για 8 επαναλήψεις αντίστοιχα και για χρόνο προσομοίωσης 3000 δευτερόλεπτα.



Σχήμα 57: Αποτελέσματα προσομοίωσης BIOMD 188 από (α) ibiosim tool (β) FRM4X για χρόνο αντίδρασης 3000 secs για την συμπεριφορά των στοιχείων, p53, Mdm2 για την βιοχημική ταλάντωση ARF.

Αντίστοιχα το σχήμα 58 (α) δείχνει την διασπορά των τιμών από την βαθμίδα FRM4X SoC για τις 8 επαναλήψεις και στο 58 (β) φαίνεται το Μέσο τετραγωνικό σφάλμα για τα στοιχεία ενδιαφέροντος.



Σχήμα 58: (α) Διασπορά τιμών των συγκεντρώσεων από τις 8 προσομοιώσεις του BIOMD 188 από FRM4X SoC για χρόνο αντίδρασης 3,000 sec τρόπο λειτουργίας ‘2’ (β) MSE ανάλυση για τις πρωτεΐνες p53, Mdm2

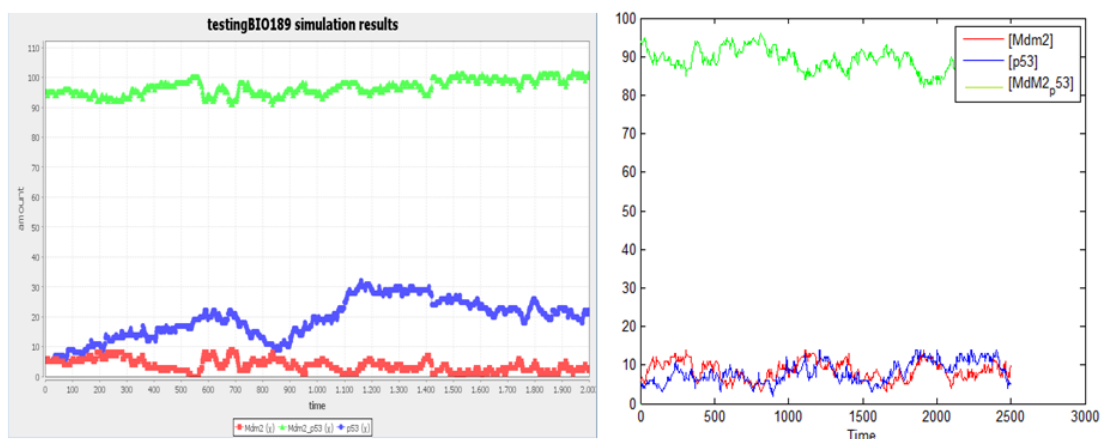
Το μέσο σφάλμα από τα στοιχεία ενδιαφέροντος είναι για την πρωτεΐνη p53 (4.54) και για την Mdm2 (11.68). Τα παρακάτω σχεδιαγράμματα είναι από τη διεθνής βιβλιογραφία όπου αναφέρουμε τις ταλαντώσεις των πρωτεϊνών p53 και Mdm2.

Για το μοντέλο αυτό χρειάστηκαν 4,000 πακέτα σε 0.056 sec για να καταγραφούν τα δεδομένα στην DDR2 του Leon στο FPGA με απόδοση 1.4M Reactions/sec. Προσμετρώντας και την εγγραφή στο δίσκο για τα 4,000 πακέτα όπου χρειαστήκαμε 1.53 δευτερόλεπτα με απόδοση 52,280 Reactions/sec.

Πίνακας 21: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel 188 για μία end to end διαδικασία.

Model: Lotka Voltera	SoC Performance			End to End performance		
	Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec
4000	0.056	1.4	0,0713	1.53	0,052	0,0026

Παρακάτω γίνεται η ανάλυση του Biomodel 189 με 19 αντιδράσεις και 18 στοιχεία στο FRM4X σε τρόπο λειτουργίας ‘2’ για 4 διαφορετικές προσομοιώσεις με χρόνο προσομοίωσης ορισμένο στα 2500 δευτερόλεπτα όπου ζητήθηκαν 4,000 πακέτα που περιγράφουν και τις 8 προσομοιώσεις FMR αλγορίθμου. Αντίστοιχα Στο σχήμα 59 (α) φαίνεται η αντίστοιχη προσομοίωση από το Ibiosim tool.



(α)

(β)

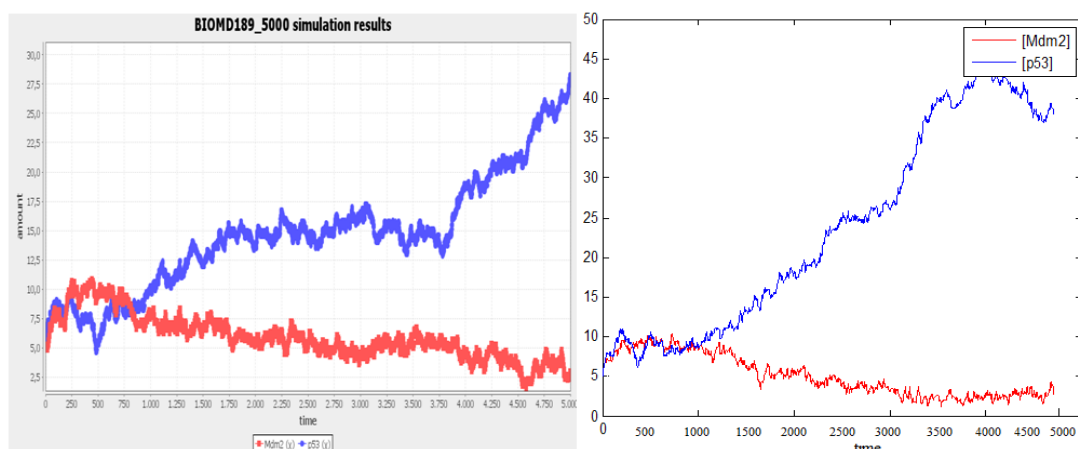
Σχήμα 59: (α) Αποτελέσματα προσομοίωσης BIOMD 189 από το FRM4X SoC για χρόνο αντίδρασης 2500 sec όπου φαίνεται η συμπεριφορά των στοιχείων Mdm2,p53 για το ATM μοντέλο από την προσομοίωση στο Ibiosim Tool. Ενώ στο (β) φαίνονται τα αποτελέσματα από την αντίστοιχη προσομοίωση στο FRM4X SoC σε τρόπο λειτουργίας'2'.

Τα πακέτα που πάρθηκαν ήταν 4,000 σε 0.056 sec στην DDR2 Leon με απόδοση 1.35M Reactions/sec. Ο χρόνος για την εγγραφή των αποτελεσμάτων στο δίσκο είναι για τα 4,000 πακέτα 1.53 seconds με απόδοση 49,666 Reactions/sec.

Πίνακας 22 Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel 189 για μία end to end διαδικασία.

Model: BIOMD189	SoC Performance			End to End performance		
Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec	MRC/sec
4000	0.056	1.4	0,074	1.53	0,049	0,00257

Το μοντέλο 189 ελέγχθηκε και στη βαθμίδα NRM4X σε τρόπο λειτουργίας '2' για χρόνο 5000 δευτερόλεπτα για 8 επαναλήψεις. Το σχήμα 60 δείχνει αντίστοιχα τις προσομοιώσεις από το Ibiosim tool για 8 επαναλήψεις από τη βαθμίδα NRM4X.

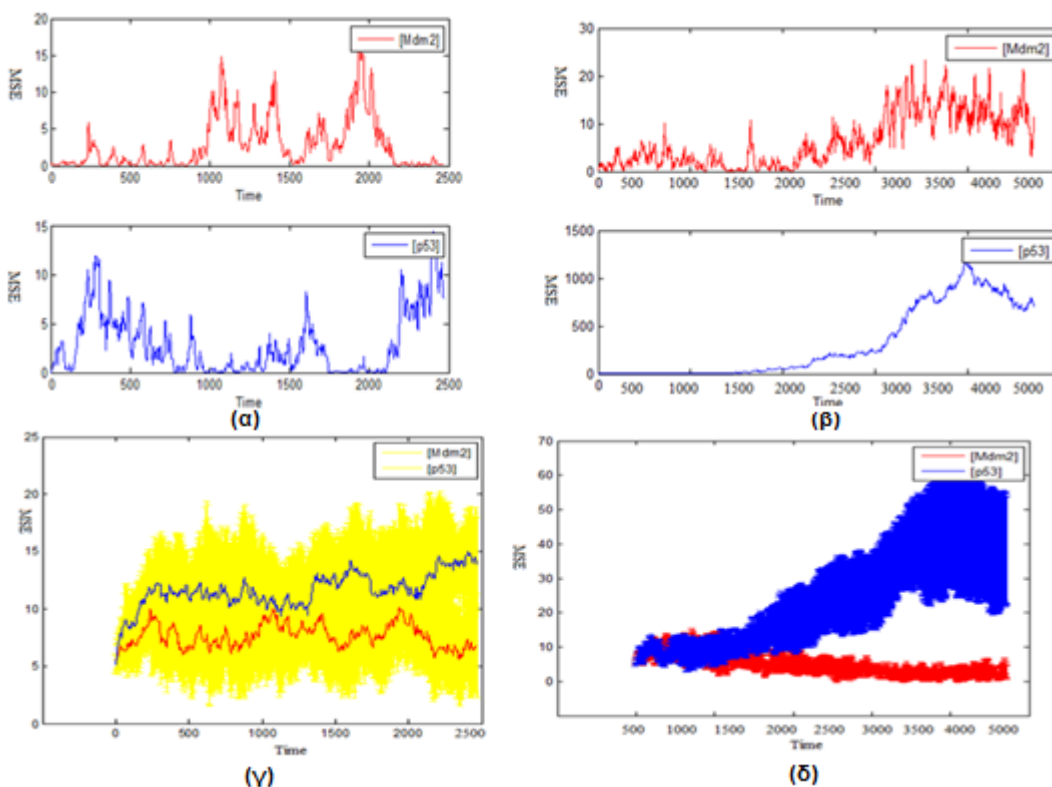


(α)

(β)

Σχήμα 60: (α) Αποτελέσματα προσομοίωσης για το Biomodel 189 με χρόνο προσομοίωσης στα 5000 seconds για 8 FRM επαναλήψεις (β) αποτελέσματα από το NRM4X SoC σε τρόπο λειτουργίας'2' για 5000 second για 8 επαναλήψεις .

Για το μοντέλο 189 βλέπουμε στο (σχήμα 61) τόσο για την FRM προσομοίωση αλλά και για την NRM τα σφάλματα και τις διακυμάνσεις των τιμών των προσομοιωτών στο FPGA για όλες τις επαναλήψεις.



Σχήμα 61: Στο (α) και στο (β) παρατηρούμε το Μέσο Τετραγωνικό Σφάλμα MSE για το μοντέλο 189 για τους προσομοιωτές NRM4X και FRM4X. Στα σχήματα (γ) και (δ) βλέπουμε αντίστοιχα για το FRM4X και για το NRM4X την διασπορά στις τιμές

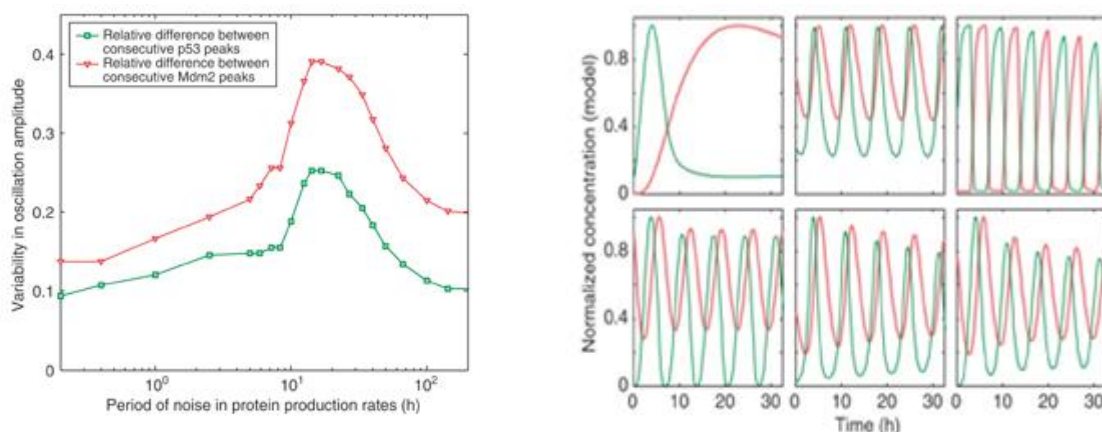
Οι πίνακες 23,24 δείχνουν τις αποδόσεις αντίστοιχα των προσομοιωτών FRM4X και NRM4X για διαδικασίες end to end για την προσομοίωση του biomodel 189 όπου και οι δύο βαθμίδες λειτουργούν σε τρόπο λειτουργίας ‘2’

Πίνακας 23: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel 189 για μία end to end διαδικασία.

Model: BIOMD189	SoC Performance			End to End performance			
	Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec	MRC/sec
	4000	0.056	1.4	0,074	1.53	0,049	0,00257

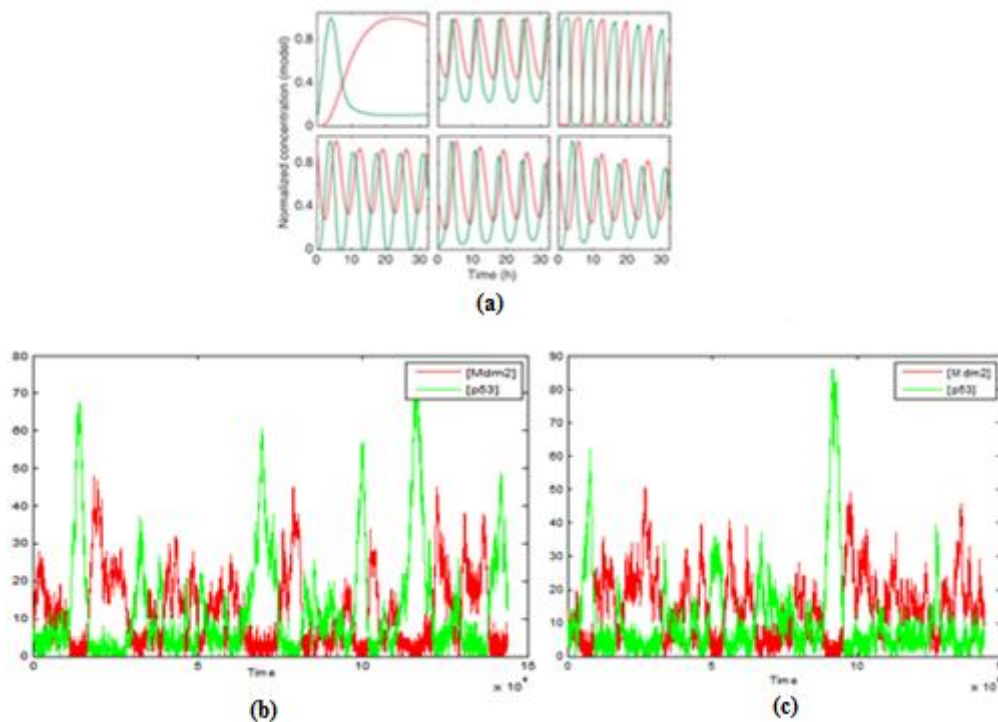
Πίνακας 24: Απόδοση συστήματος προσομοίωσης NRM4X SoC+Leon για το biomodel 189 για μία end to end διαδικασία.

Model: BIOMD189	SoC Performance			End to End performance			
	Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec	MRC/sec
	10000	0.08	2.37	0,1247	3.8	0,05	0,0026



Σχήμα 62: Αποτελέσματα από την διεθνή βιβλιογραφία από προσομοιώσεις για χρόνο αντίδρασης 30h για την συμπεριφορά των στοιχείων Mdm2,p53, για το ATM μοντέλο και το ARF αντίστοιχα.

Συγκρίνοντας τα αποτελέσματα όλων των προσομοιώσεων με στοιχεία και από την διεθνή βιβλιογραφία μπορούμε να δούμε για χρόνο 30 ωρών, τις εξής ακόλουθες προσομοιώσεις και για τα 2 Biomodels 188 και 189 από το FRM4X SoC.



Σχήμα 63: Αποτελέσματα προσομοίωσης (a) BIOMD 188 189 από βιβλιογραφία. (b),(c) αποτελέσματα από FRM4X για χρόνο αντίδρασης 144000 second 30h για την συμπεριφορά των στοιχείων Mdm2,p53, για το ATM μοντέλο και το ARF αντίστοιχα.

Με βάση τις συγκρίσεις των προσομοιώσεων του FRM4X SoC με τον προσομοιωτή Ibiosim για τα μοντέλα BIOM 188 και 189 διαπιστώθηκε ταύτιση των αποτελεσμάτων καθώς επίσης και με την σύγκριση της προσομοίωσης του FRM4X SoC για προσομοίωση των BIOMD 188 και 189 για χρόνο προσομοίωσης 30 h σε σύγκριση με τα στοιχεία της διεθνούς βιβλιογραφίας σχήμα 63 (a),(b),(c)

Πίνακας 25: Στατιστικά στοιχεία επιδόσεων (speedup) και σύγκριση των προσομοιώσεων στο FRM4X και στο Ibiosim για τα Biomodels 188 και 189.

Model = BioM188	FPGA	IBiosim	speedup
Simulation time (sec.)	0.056	7.34	131
End to End (Time/Performance)	1.53/0,052	7.34/0,011	4.79

Model=BioM189	FPGA	IBiosim	speedup
Simulation time (sec.)	0.056	7.11	127
End to End (Time/Performance)	1.53/0,049	7.11/0,011	4.64

Πίνακας 26: Στατιστικά στοιχεία επιδόσεων (speedup) και σύγκριση με FRM4X και Ibiosim για το Biomodels 189 για χρόνο προσομοίωσης 30 ωρών.

BioM189	FPGA	IBiosim	speedup
Simulation time (sec.)	0.08	21.18	265
End to End (Time/Performance)	3.8/0.05	23/0,008	6

7.6 Μοντέλο περιγραφής της λειτουργίας της πρωτεΐνης Asyn στην νόσο του Parkinson

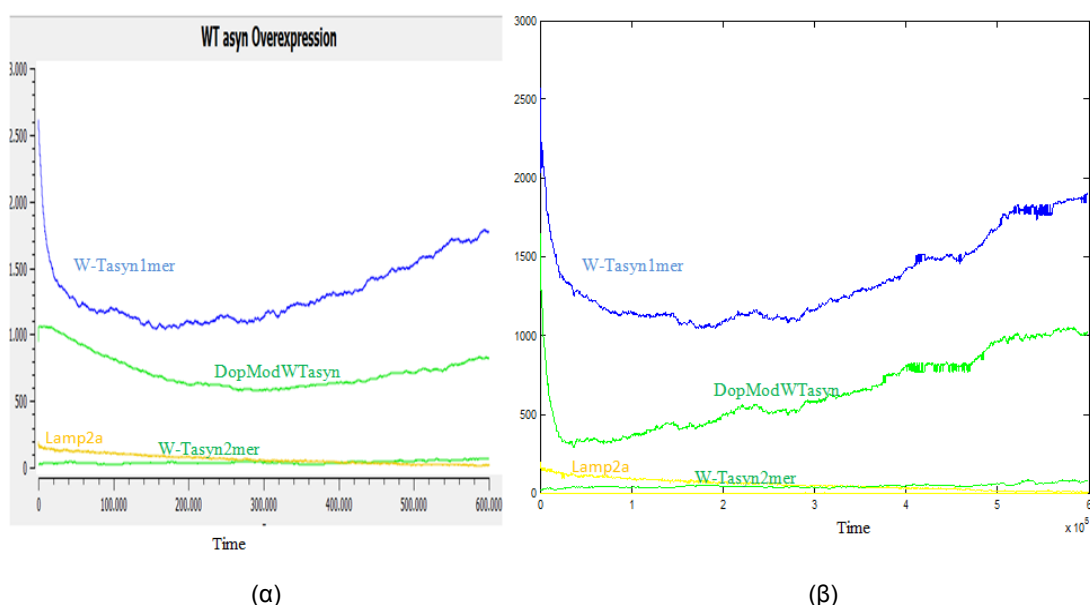
Το επόμενο μοντέλο που ελέγχθηκε αφορά το ρόλο της πρωτεΐνης α-συνουκλείνη στη νόσο του Parkinson. Στον εγκέφαλο των ασθενών με Πάρκινσον, υπάρχει απώλεια νευρώνων σε μια σημαντική λειτουργική δομή (μέλαινα ουσία) η οποία ελέγχει τη μυϊκή κίνηση. Οι νευρώνες αυτοί παράγουν την ουσία ντοπαμίνη που έχει ρόλο νευροδιαβιβαστή. Η ντοπαμίνη έχει καθοριστικό ρόλο στο συντονισμό των κινήσεων. Μεταφέρει τα νευρικά μηνύματα μεταξύ της μέλαινας ουσίας και του ραβδωτού σώματος του εγκεφάλου που έχει βασικό ρόλο στην ομαλότητα και στο συντονισμό και έλεγχο των κινήσεων. Κανένας δεν γνωρίζει τι είναι εκείνο που οδηγεί στη σταδιακή και προοδευτική καταστροφή των νευρώνων της μέλαινας ουσίας στον εγκέφαλο. Όλοι οι άνθρωποι, με την αύξηση της ηλικίας, χάνουν κάποιο αριθμό νευρώνων της μέλαινας ουσίας που παράγει την ντοπαμίνη. Όμως στους ασθενείς με Πάρκινσον παρατηρείται απώλεια του 50% ή περισσότερο των νευρώνων της μέλαινας ουσίας. Ο ρόλος των γονιδίων στην παθογένεια της ασθένειας δεν έχει ακόμη πλήρως διαλευκανθεί. Βασικά αναγνωρίστηκαν δύο γονίδια που παρουσιάζουν ανωμαλίες, και ευθύνονται για τη σύνθεση της α-συνουκλείνης. Σε ασθενείς με Πάρκινσον παράγεται η εν λόγω πρωτεΐνη με αλλοιώσεις. Το δεύτερο γονίδιο εμπλέκεται σε διαταραχές του μηχανισμού με τον οποίο ο οργανισμός αποβάλλει τις άχρηστες πρωτεΐνες.

Το συγκεκριμένο μοντέλο αποτελείται από 136 αντιδράσεις και 93 στοιχεία. Αυτό που χαρακτηρίζει το μοντέλο είναι ότι χρειάζεται 7 εργαστηριακές μέρες για να διαπιστώσουμε το φάσμα των μεταβολών της πρωτεΐνης της α-συνουκλεάσης και των υπό-προϊόντων της. Δηλαδή η προσομοίωση χαρακτηρίζεται από έναν μεγάλο όγκο δεδομένων για τη κάθε

προσομοίωση μιας και ο χρόνος προσομοίωσης παράμετρος (T_{sim}) ανέρχεται στα 600,000 sec. Το μοντέλο προσομοιώθηκε στο FRM4X έγιναν πειράματα και με τους 2 τρόπους λειτουργίας και πιο συγκεκριμένα οι κυματομορφές οι οποίες πάρθηκαν είναι για τρόπο λειτουργίας $k=0$ δηλαδή μία N παράλληλη προσομοίωση και για 8 επαναλήψεις.

Η ασθένεια του Parkinson συνοδεύεται από συσσωματώματα Lewy τα οποία δημιουργούνται από τον πολυμερισμό της α_{syn} σε ολιγομερή. Αυτό συμβαίνει λόγω λαθών (μεταλλάξεων) του γονιδίου που δημιουργεί την α_{syn} του γονιδίου (SNCA). Λόγω των λαθών αυτών δημιουργούνται οι πρωτεΐνες α_{syn} οι οποίες αποδομούνται (πολυμερισμό) σε ολιγομερή τα οποία και παρατηρούμε με βάση την προσομοίωση τού σχήματος 64, τα οποία όμως δεν αποδομούνται από τον οργανισμό με αποτέλεσμα την συσσώρευση τους και την συμβολή στη δημιουργία των συσσωματωμάτων Lewy τα οποία βλάπτουν τις νευρικές απολήξεις.

Τα παρακάτω διαγράμματα δείχνουν τις προσομοιώσεις από το Corasi σε σύγκριση με την προσομοίωση του FRM4X (σχήμα 64) για την συμπεριφορά συγκεκριμένων στοιχείων μονομερών που παρατηρούνται όπως επίσης και την εξέλιξη του φάσματος της α -συνουκλείνης και της Ντοπαμίνης και των άλλων στοιχείων που λαμβάνουν χώρα στις αντιδράσεις αυτές.



Σχήμα 64: Παρατηρούμε στο σχήμα (α) τις μέσες τιμές των συγκεντρώσεων των στοιχείων ενδιαφέροντος από τον προσομοιωτή Corasi. Στο (β) αντίστοιχα παρατηρούμε τις μέσες τιμές του FRM4X SoC σε τρόπο λειτουργίας '1' για 8 επαναλήψεις του FRM αλγορίθμου ορίζοντας τον χρόνο προσομοίωσης στα 600,000 δευτερόλεπτα.

Οι παρακάτω καμπύλες δείχνουν το μέσο συστηματικό σφάλμα που παρατηρείτε σε όλο το εύρος της κυματομορφής με βάση την εξίσωση 26. Επίσης αναδεικνύουμε την

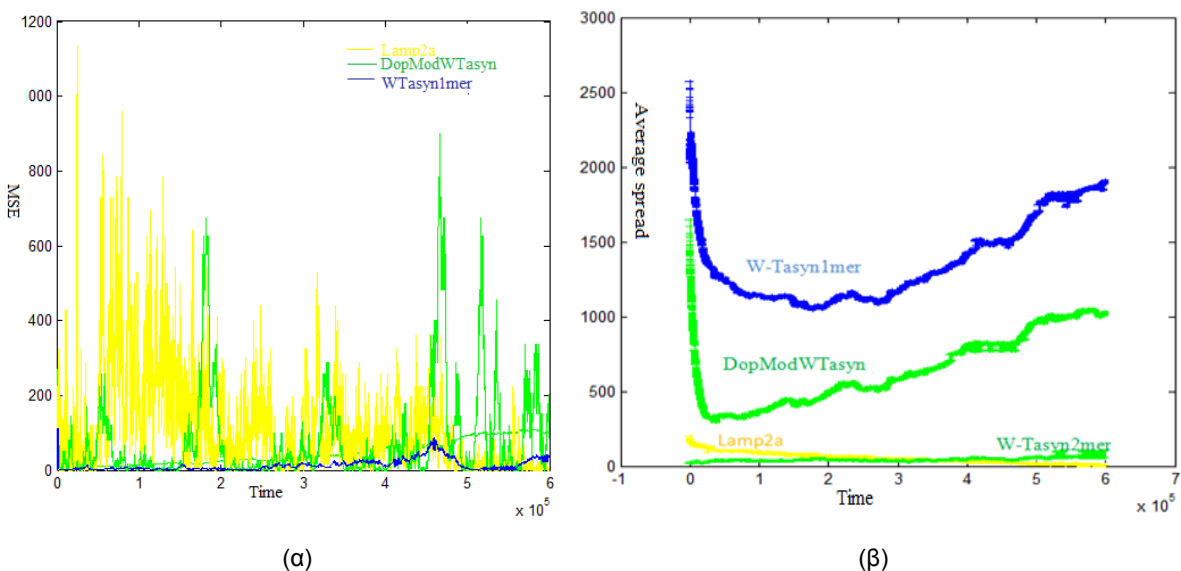
διασπορά τιμών (αποκλίσεις τιμών από τις 8 επαναλήψεις) για τα στοιχεία ενδιαφέροντος (σχήμα 65). Ο πίνακας 27 δείχνει αντίστοιχα το Μέσο τετραγωνικό σφάλμα που παρατηρείται για τις 8 επαναλήψεις για όλο το φάσμα των κυματομορφών.

Γενικά όταν η α-συνουκλείνη υπέρ-εκφράζεται, τα ολιγομερή (προϊόντα λόγω του πολυμερισμού) δεν μπορούν να αποδομηθούν από τον παράγοντα CMA, όμως μπορούν και προσκολλούνται στον παράγων Lamp2a (lysosomal receptor) και άρα ο παράγων αυτός δρα κατασταλτικά. Υπάρχει όμως και ένας άλλος μηχανισμός όπου η Ντοπαμίνη αντιδρά με την α-συνουκλεάση και το νέο προϊόν αντιδρά με το Lamp2a με αποτέλεσμα την αναστολή του παράγων Lamp2a και την ύπερ έκφραση της α-συν.

Επίσης η συνεχής συγκέντρωση ολιγομερών αποδεικνύεται βλαβερή για την διατήρηση της ομοιόστασης του πρωτεασώματος.

Όλες οι προαναφερθέντες ουσίες υπάρχουν και συμμετέχουν σε βιοχημικές αντιδράσεις των νευρικών κυττάρων. Οι μεταλλάξεις που ενδεχομένως συμβαίνουν λόγω βλάβης στην αλληλουχία των γονιδίων παραγωγής της α-συν δημιουργεί αστάθεια στις ουσίες αυτές με αποτέλεσμα να υπάρχουν όλες οι βλαβερές συνέπειες για το νευρικό κύτταρο που αναφέραμε προηγουμένως, με αποτέλεσμα την καταστροφή του κυττάρου.

Παρακάτω φαίνονται οι προσομοιώσεις για τα στοιχεία ενδιαφέροντος από το FRM4X SoC



Σχήμα 65: (α) Μέσο συστηματικό σφάλμα για τα στοιχεία ενδιαφέροντος (παράγοντας Lamp2a, μονομερές Wild Type asyn και Dopamine Wild Type asyn) (β) Διασπορά τιμών συγκεντρώσεων από το FRM4X SoC για 8 επαναλήψεις.

Πίνακας 27: Μέσο τετραγωνικό σφάλμα για τα στοιχεία ενδιαφέροντος.

Species	MSE
DopModWtasyn	46.71
Lamp2a	110.9
WTasyn2mer	80.136
WTasyn1mer	13.33

Αντίστοιχα ο πίνακας 28 δείχνει τις αποδόσεις για το σύστημα FRM4X SoC και Leon. Σε όλους του πίνακες απόδοσης δείχνουμε το χρόνο και αντίστοιχα την απόδοση των συστημάτων, αρχικά μέχρι τα στοιχεία να καταγραφούν στην μνήμη DDR2 του FPGA και έπειτα τον χρόνο μέχρι να εισαχθούν στο τερματικό μέσο της θύρας Ethernet.

Πίνακας 28: Απόδοση συστήματος προσομοίωσης FRM4X SoC+Leon για το biomodel Asyn για μία end to end διαδικασία.

Model: Asyn	SoC Performance			End to End performance		
Reaction Cycles	Time (sec)	MR/sec	MRC/sec	Time (sec)	MR/sec	MRC/sec
135000	1.89	9.7	0.071	51.3	0,36	0,0026

Οι πίνακες 29 και 30 δείχνουν τις συγκρίσεις σε χρόνο προσομοίωσης της πλατφόρμας FRM4X και Leon με τον γνωστό προσομοιωτή Copasi προσομοιώνοντας το μοντέλο της Asyn. Αντίστοιχα παρατηρούμε τους μεγάλους χρόνους επιτάχυνσης που παρατηρούνται. Η αύξηση της επιτάχυνσης και αντίστοιχα της απόδοσης σε σχέση με όλους τους Software προσομοιωτές αυξάνεται καθώς αυξάνετε η πολυπλοκότητα των μοντέλων.

Πίνακας 29: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση του FRM4X με το Ibiosim για το Biomodel Asyn.

Asyn_Defence	FPGA	IBiosim	speedup
Simulation time (sec.)	1.89	16164	8552
End to End Time(sec)/Performance (MR/sec)	60.5**/ 0.36	16164/0,0011	267

7.7 Μοντέλο περιγραφής της λειτουργίας του ενζύμου UCHL1 στην νόσο του Parkinson.

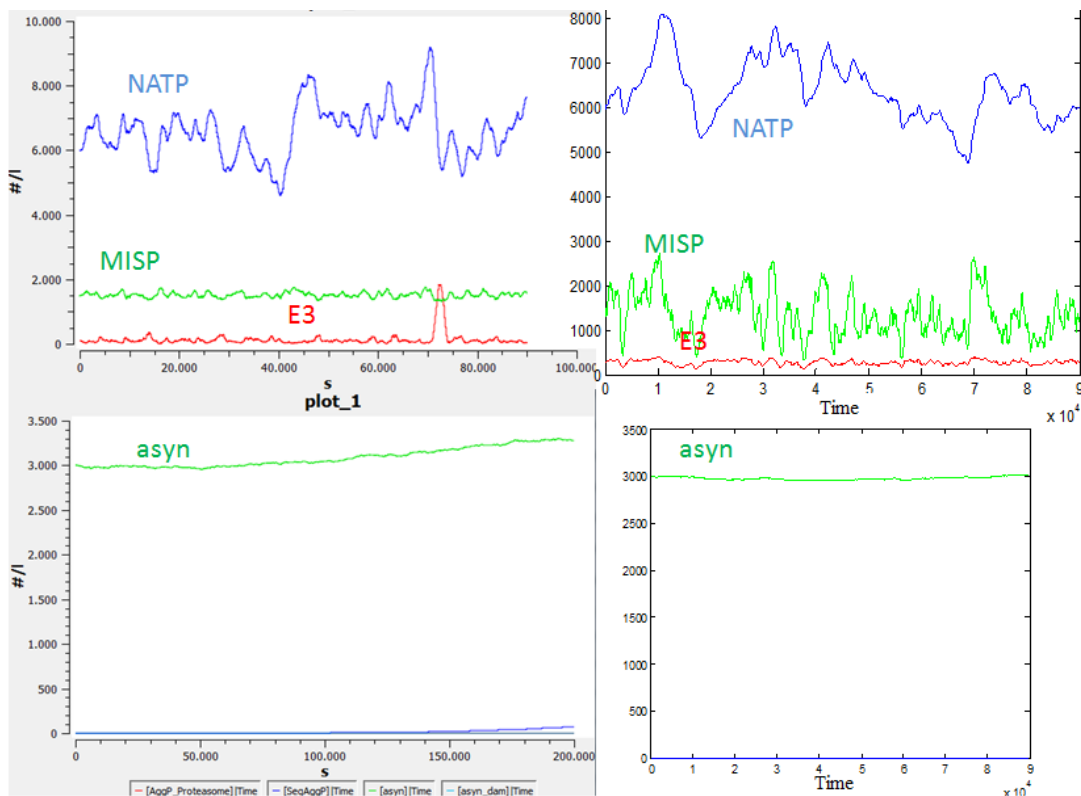
Το μοντέλο αυτό θεωρείται ως ένα από τα πιο σύνθετα μοντέλα καθώς αποτελείται από 316 αντιδράσεις και 136 στοιχεία. Πιο συγκεκριμένα αναλύεται ο ρόλος του ενζύμου UCHL1 στη μετάλλαξη του γονιδίου που δημιουργεί την πρωτεΐνη Asyn.

Έξι γονίδια έχουν ενοχοποιηθεί για την νόσο αυτή (ά-synuclein, parkin, UCH-L1, DJ-1, PINK1 και LRRK2). Οι πρωτεΐνες που κωδικοποιούνται από τα 3 πρώτα γονίδια έχουν άμεση σχέση με την ομοιόσταση του πρωτεασώματος όπου παίζει χαρακτηριστικό ρόλο

στην εξέλιξη της νόσου. Εμείς θα ασχοληθούμε με την δημιουργία του ενζύμου UCHL1 και τον καταλειπτικό του ρόλο στην μετάλαξη του γονιδίου που παράγει την πρωτεΐνη A β μέσω ενός πολύπλοκου βιοχημικού μηχανισμού.

Αυτό το μοντέλο προσομοιώθηκε και με τους 2 αλγόριθμους (NRM/FRM) στην πλατφόρμα FRM4X_NRM4X. Αρχικά λοιπόν θα δείξουμε τις προσομοιώσεις από το FRM4X και ακολούθως για το NRM4X σε τρόπο λειτουργίας '1' (για μία παράλληλη προσομοίωση) και για τους 2 αλγόριθμους καθώς σε αυτό το μοντέλο φαίνεται έντονα η ανάγκη παραλληλοποίησης για αύξηση της απόδοσης καθώς το μοντέλο περιλαμβάνει 316 αντιδράσεις και 136 στοιχεία. Το σχήμα 66 δείχνει τα αποτελέσματα από την προσομοίωση στο FRM4X για χρόνο 90,000 δευτερόλεπτα αντίστοιχα από το Ibiosim Tool για 8 επαναλήψεις του FMR αλγόριθμου. Αντίστοιχα στο σχήμα 67 παρουσιάζουμε τα αποτελέσματα από την προσομοίωση με τον NRM και τον FRM αλγόριθμο καθώς και την σύγκριση τους με τα αντίστοιχα γραφήματα προσομοίωσης του μοντέλου από την βιβλιογραφία όπου παρατηρείται ταύτιση των τιμών.

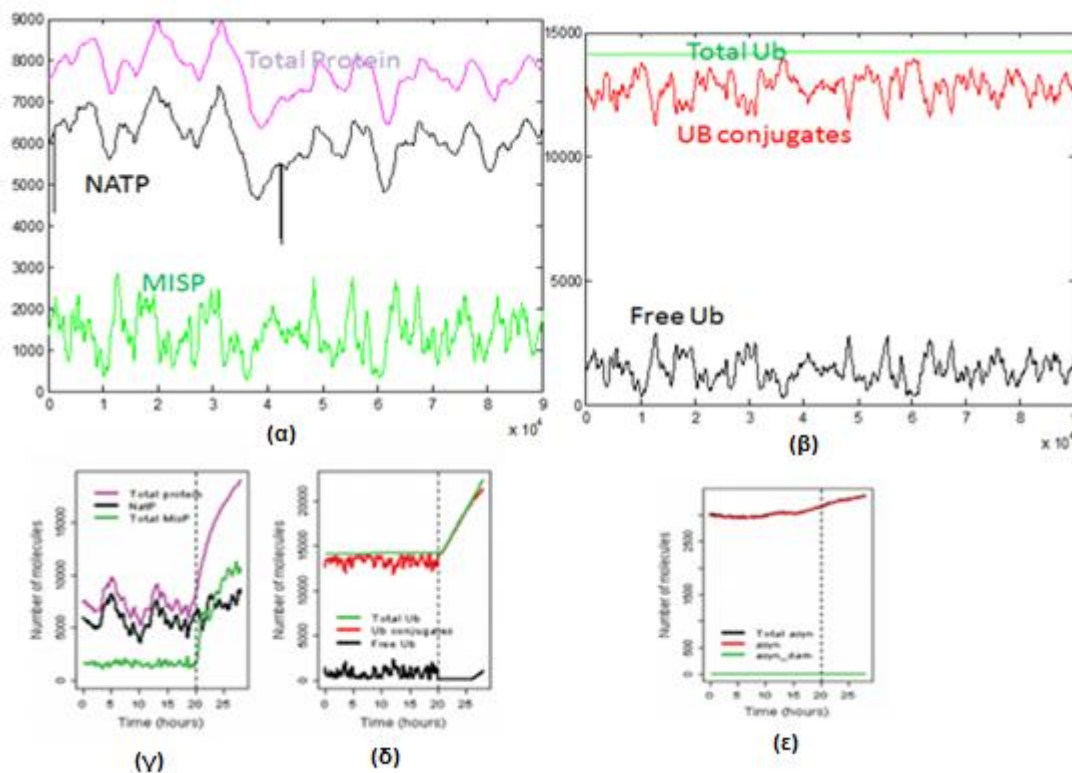
Από την ανάλυση του μοντέλου στο Copasi και αντίστοιχα στους προσομοιωτές FRM4X και NRM4X προκύπτουν τα αντίστοιχα διαγράμματα για το μέσο τετραγωνικό σφάλμα MSE καθώς και για την διασπορά των τιμών (σχήμα 68) όπως αναφέρονται και στο πίνακα 30.



(α)

(β)

Σχήμα 66: (α) Αποτελέσματα προσομοίωσης από το Ibiosim για 8 διαφορετικές προσομοιώσεις για χρόνο προσομοίωσης 90,000 sec, (β) Αποτελέσματα από την προσομοίωση του FRM αλγορίθμου 8 επαναλήψεις, χρόνος προσομοίωσης 90,000.



(α)

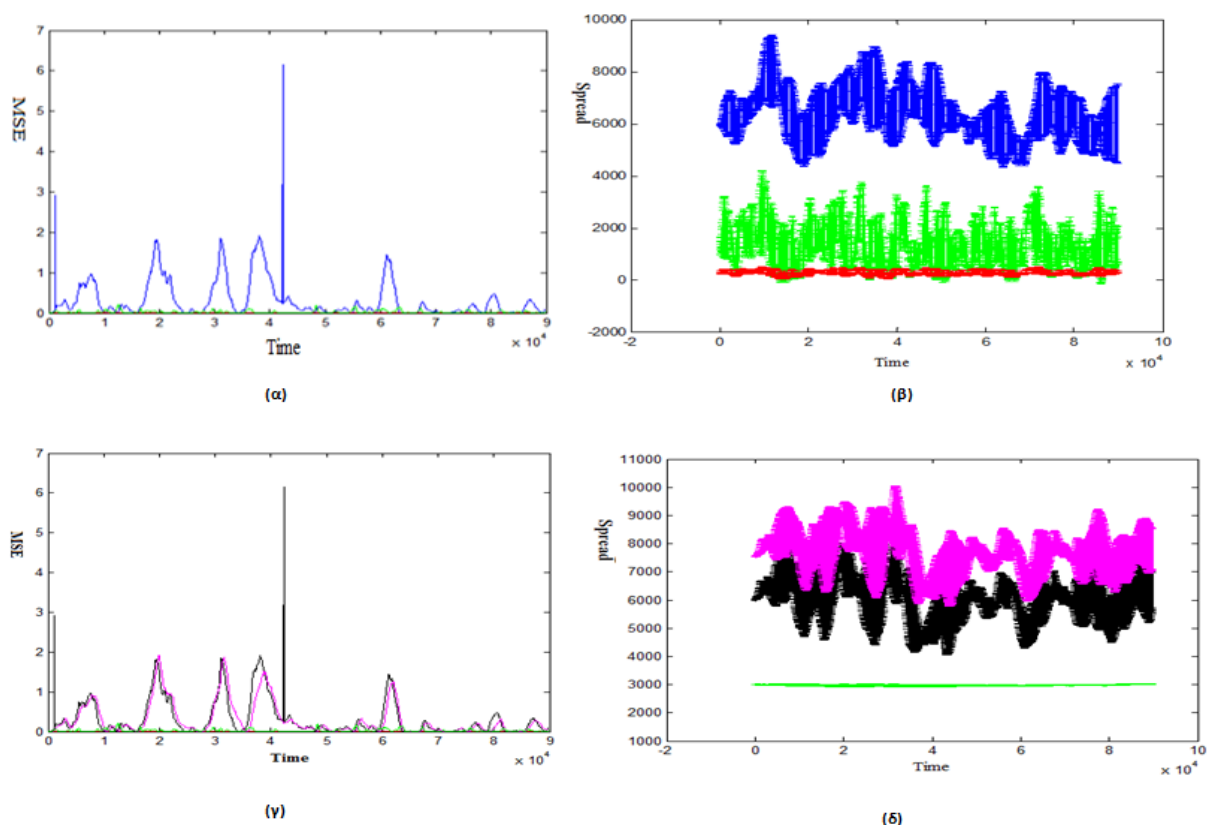
(β)

(γ)

(δ)

(ε)

Σχήμα 67: Αποτελέσματα προσομοίωσης στοιχείων ενδιαφέροντος (α) για τις προσομοιώσεις στο FRM4X και (β) για τον προσομοιωτή NRM4X σε τρόπο λειτουργίας '1' με χρόνο προσομοίωσης 90,000 seconds για 8 επαναλήψεις. Ακολούθως τα διαγράμματα (γ)(δ) και (ε) δείχνουν τα αποτελέσματα αντίστοιχων προσομοιώσεων για το UCHL1 ND σύστημα



Σχήμα 68: Αποτελέσματα προσομοίωσης στοιχείων ενδιαφέροντος (α) Μέσο συστηματικό σφάλμα για τον αλγόριθμο FRM για τις μέσες τιμές από το Copasi συγκρινόμενες με τις τιμές του FRM4X SoC. (β) Η αντίστοιχη διακύμανση τιμών από τις 8 προσομοιώσεις του FRM4X σε τρόπο λειτουργίας ‘1’ για χρόνο προσομοίωσης 90,000 second. Ακολούθως το διάγραμμα (γ) δείχνει το MSE για τις προσομοιώσεις στο NRM4X συγκρινόμενες με τις προσομοιώσεις στο Copasi και αντίστοιχα στο (δ) φαίνεται η διακύμανση των τιμών για την NRM προσομοίωση.

Πίνακας 30: Μέσο τετραγωνικό σφάλμα για τα στοιχεία ενδιαφέροντος από τους προσομοιωτές NRM4X και FRM4X για την προσομοίωση του μοντέλου UCHL1.

Species	MSE NRM4X	MSE FRM4X
NATP	3.2	6
MISP	2.7	2.5
Asyn	12,7	13,71
Lamp2a	2,33	0,3
UCHL1	2.59	0,11

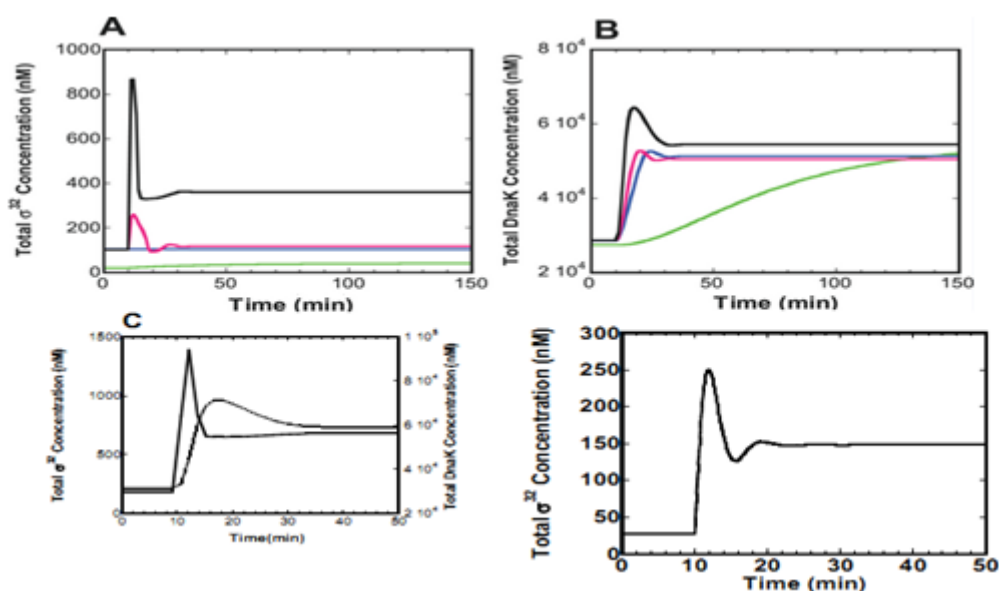
7.8 Προσομοίωση μοντέλων στις μικρότερης κλίμακας προσομοιωτές FRM1X/FRM2X .

7.8.1 Προσομοίωση μοντέλου E-Coli.

Το επόμενο μοντέλο προς προσομοίωση είναι το Heat Response Shock of E-Coli. Αυτό το μοντέλο αποτελείται από 61 αντιδράσεις και 29 στοιχεία. Γενικά στα κύτταρα υπάρχουν κάποιοι μηχανισμοί αντίδρασης που έχουν σχέση με την ομοιόσταση των οργανισμών. Σε αλλαγές θερμοκρασίας ή σε περιπτώσεις έκθεσης σε ακτινοβολία εμφανίζονται από

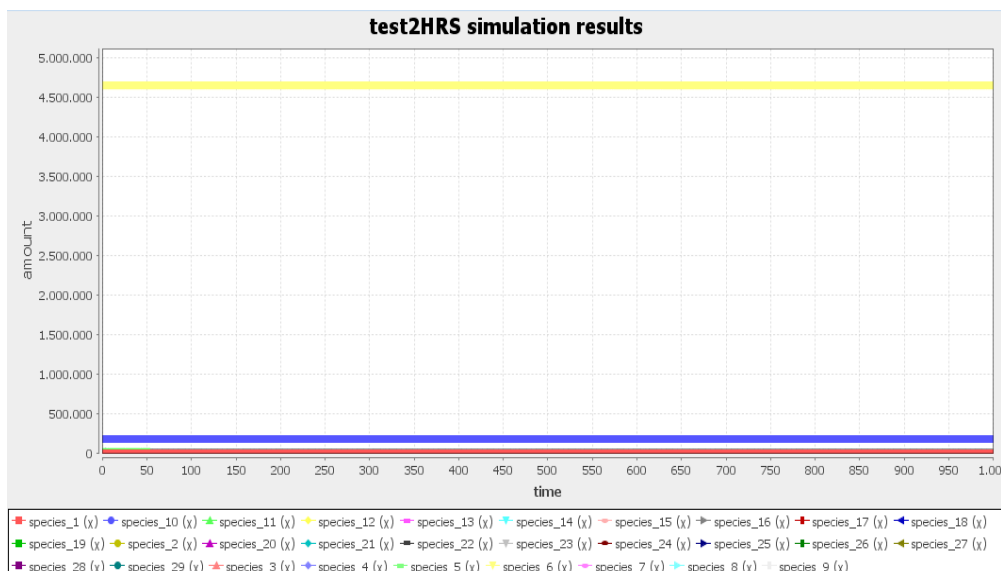
μεταβολικές αντιδράσεις κάποια υπό προϊόντα που συμβάλουν στην παραγωγή πρωτεϊνών γνωστές ως (heat shock proteins). Αυτός ο μηχανισμός συμβαίνει σε όλους τους έμβιους οργανισμούς. Οι ερευνητές μελέτησαν αυτό το μοντέλο στα βακτήρια E-Coli. Τα E-Coli βακτήρια βρίσκονται σχεδόν παντού και είναι εύκολη η μελέτη τους. Οι πρωτεΐνες (Heat Shock) εμφανίζουν αύξηση σε μια αλλαγή θερμοκρασίας από 30 °C σε 42 °C. Αυτές οι πρωτεΐνες εμφανίζουν συνδεδεμένες στο μακρομόριο τους ένα παράγοντα $E\sigma^{32}$ ο οποίος αναγνωρίζεται από την RNA πολυμεράση. Αυτό που μελετάται είναι η σταθεροποίηση των μηχανισμών των βακτηριδίων E-Coli για την σταθεροποίηση των Heat Shock πρωτεϊνών που παίζουν ιδιαίτερα σημαντικό ρόλο στην επιβίωση των οργανισμών. Επίσης μια άλλη ενδιαφέρουσα έρευνα είναι ο ρόλος αυτών των πρωτεϊνών στην εξέλιξη του καρκίνου.

Τα παρακάτω σχεδιαγράμματα προέρχονται από προσομοιώσεις αντίστοιχων μοντέλων HRS. Παρατηρούμαι ότι υπάρχει μια στιγμιαία περιοχή αστάθειας λόγω απότομης θερμοκρασιακής αλλαγής. Όμως σχεδόν πάντα όπως παρατηρούμαι επέρχεται ισορροπία.



Σχήμα 69: Αποτελέσματα προσομοίωσης in Vitro για το Heat response of E-Coli όπου φαίνονται απότομες μεταβολές θερμοκρασίας και σταθεροποίηση.

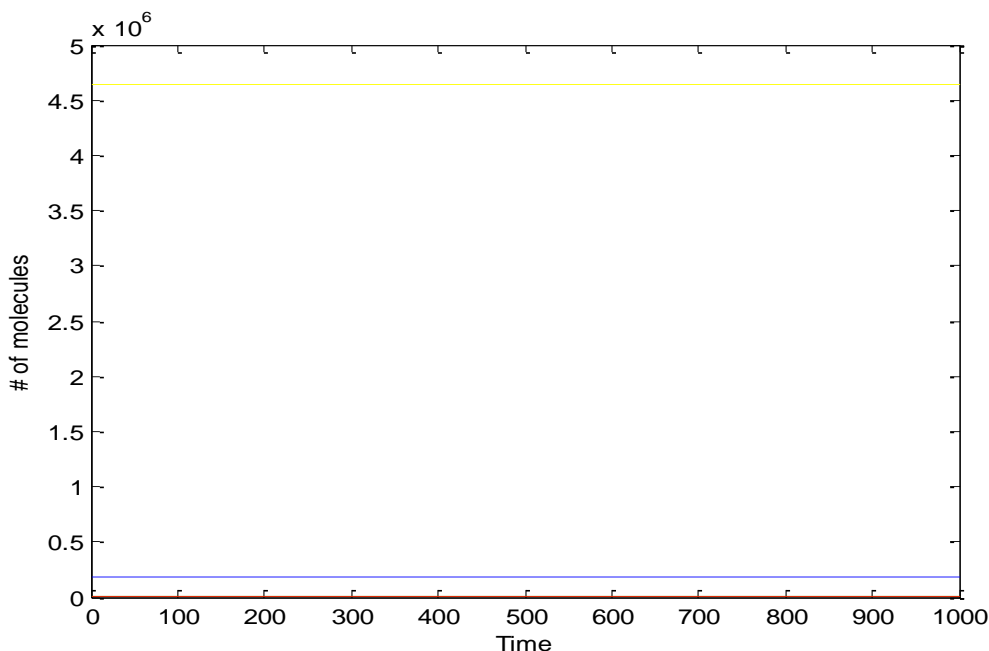
Τα παρακάτω σχεδιάγραμμα είναι από προσομοιώσεις του HRS Biomodel στο Ibiosim tool. Παρατηρούμαι από τα σχεδιαγράμματα ότι το σύστημα μας είναι πάντα σε ισορροπία. Αυτό σημαίνει ότι δεν υπάρχουν θερμοκρασιακές αλλαγές ή ότι οι μηχανισμοί σταθεροποίησης δεν αφήνουν το σύστημα να επέλθει σε αστάθεια δηλαδή οι πρωτεΐνες με το παράγων $E\sigma^{32}$ παραμένουν σε σταθερή κατάσταση.



Σχήμα 70: Αποτελέσματα προσομοιώσεων HRS από iBiosim tool για χρόνο αντίδρασης 1000 δευτερολέπτων συμπεριφορά στοιχείων X1,X5,X6 καθώς και για όλα τα στοιχεία.

Το μοντέλο αυτό προσομοιώθηκε σε όλες τις βαθμίδες FRM SoC καθώς και στην FRM4X_NRM4X όπου έγινε και έλεγχος της λειτουργίας του NRM. Επίσης και σε αυτό το μοντέλο ελέγχθηκαν όλοι οι τρόποι λειτουργίας για $k=1$ και για $k=0$.

Τα αποτελέσματα των παρακάτω κυματομορφών είναι από την προσομοίωση με το FRM2X SoC σε τρόπο λειτουργίας $k=0$ δηλαδή για μία N παράλληλη προσομοίωση. Στην οποία χρειάστηκαν 80,000 πακέτα για 8 προσομοιώσεις (runs).



Σχήμα 71: Αποτελέσματα προσομοίωσης HRS από FRM2X SoC για χρόνο αντίδρασης 1000 second συμπεριφορά στοιχείων X1,X6,X18,X10 σε τρόπο λειτουργίας '1'.

Για τις κυματομορφές χρειάστηκαν 80,000 πακέτα σε 1.12 sec στην DDR2 Leon και η απόδοση ανήλθε στα 4.8M Reactions/sec. Ο χρόνος εγγραφής στο δίσκο ανήλθε στα 30.7 sec με απόδοση της τάξης των 158,950 Reactions/sec.

Στο μοντέλο αυτό παρατηρήσαμε την σταθερότητα στους πληθυσμούς των στοιχείων πράγμα το οποίο γίνεται λόγω των μηχανισμών ανάδρασης για την διατήρηση της ομοιότητας στους οργανισμούς. Παρακάτω στον πίνακα 31 παρατηρούμαι την σύγκριση ως προς το speedup για το FRM4X και το Ibiosim tool.

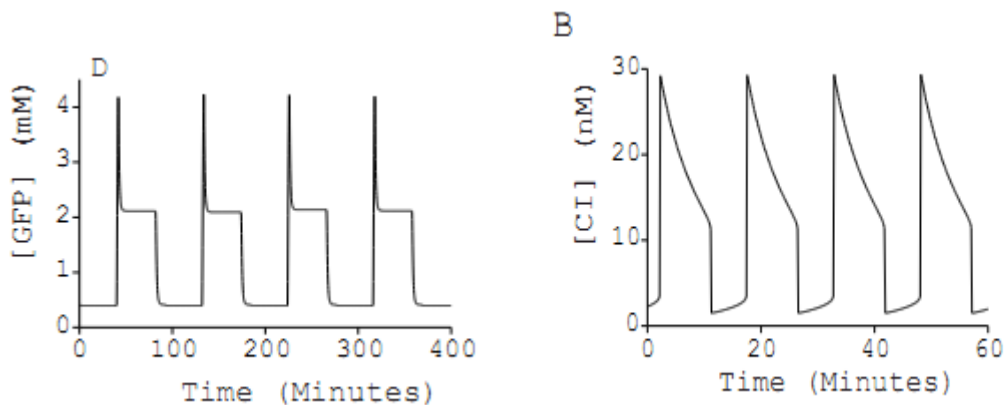
Πίνακας 31: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση με FRM2X και Ibiosim για το HRS biomodel.

HRS	FPGA	iBiosim	speedup
Simulation time (sec.)	1.12	9000	8035x
End to End (including writing data from Ethernet to disk)	30.7	9000	293x

7.8.2 Προσομοίωση του Autoregulatory Gene Network Biomodel

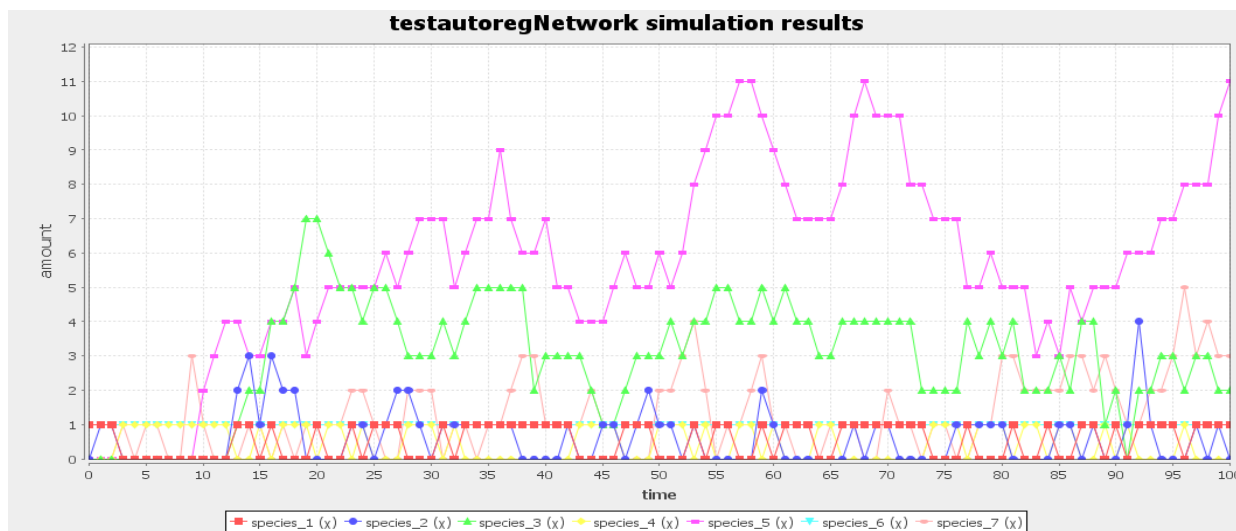
Το επόμενο μοντέλο που ελέγχθηκε είναι πάλι από την Biomodels Database και αφορά αντιδράσεις για την παραγωγή πρωτεϊνών. Το συγκεκριμένο μοντέλο αποτελείται από 10 αντιδράσεις των 7 στοιχείων. Ουσιαστικά περιγράφεται ο τρόπος με τον οποίο παράγονται οι πρωτεΐνες από την γονιδιακή αλληλουχία των βάσεων στο DNA, την μεταφορά αυτής της πληροφορίας μέσω του mRNA στα ριβοσώματα με απώτερο σκοπό την παραγωγή των πρωτεϊνών. Ουσιαστικά εδώ μελετάται το σύστημα ελέγχου παραγωγής και καταστολής των πρωτεϊνών ώστε το σύστημα μας να βρίσκεται πάντα σε ισορροπία. Βρέθηκε λοιπόν ότι όταν παράγονται από την συγκεκριμένη αλληλουχία βάσεων οι πρωτεΐνες στα ριβοσώματα ο υπερπληθυσμός αυτών οδηγεί το σύστημα σε περεταίρω αντιδράσεις και την παραγωγή κάποιου ενδιάμεσου προϊόντος το οποίο όμως προκύπτει ότι είναι και μηχανισμός ελέγχου διότι το mRNA αντιδρά με αυτό το υποπροϊόν και αναστέλλει την διαδικασία παραγωγής περεταίρω πρωτεϊνών.

Παρακάτω φαίνονται οι κυματομορφές με ανάλογα συστήματα γονιδιακών αυτοελεγχόμενων μηχανισμών παραγωγής πρωτεϊνών.



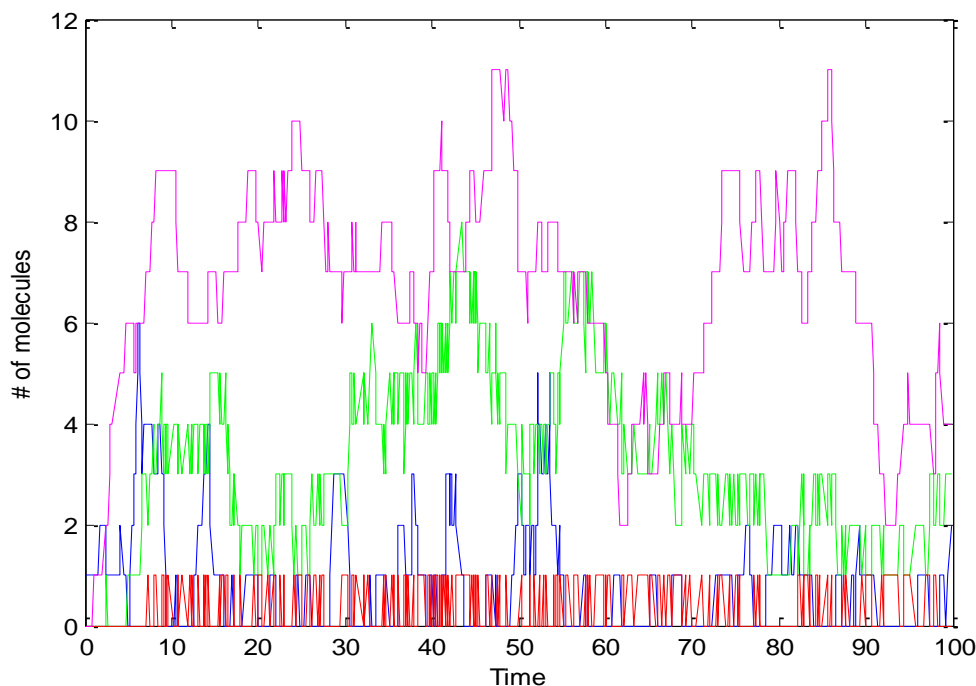
Σχήμα 72: Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων από την διεθνή βιβλιογραφία για την παραγωγή των πρωτεϊνών GFP και CT.

Το παρακάτω διάγραμμα δείχνει την προσομοίωση του μοντέλου για 10 διαφορετικές προσομοιώσεις του μοντέλου από το προσομοιωτή Ibiosim Tool.



Σχήμα 73 : Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων από Ibiosim tool για χρόνο αντίδρασης 100 secs για την συμπεριφορά των στοιχείων 1x-7x.

Το μοντέλο ελέγχθηκε στο FRM4X SoC σε τρόπο λειτουργίας $k=1$ όπου έγιναν 10 διαφορετικές προσομοιώσεις και χρειάστηκαν 7,500 πακέτα. Τα αποτελέσματα φαίνονται στις παρακάτω κυματομορφές.



Σχήμα 74: Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων από το FRM4X SoC για χρόνο αντίδρασης 100 sec .

Τα πακέτα που λήφθηκαν είναι 7,500 σε 0.1 sec στην DDR2 Leon με απόδοση 52500 Reactions/sec. Ο χρόνος καταγραφής στο δίσκο για τα 7500 ανήλθε στα 2.88 sec με απόδοση 18229 Reactions.

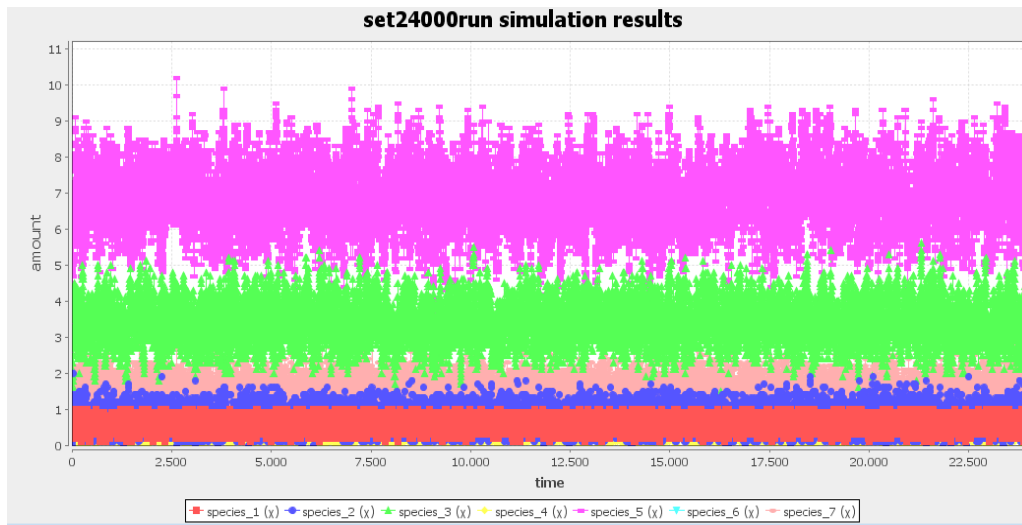
Πίνακας 32: Στατιστικά στοιχεία επιδόσεων (speedup) σε σύγκριση με FRM4X και Ibiosim για το Autoregulatory Gene Network biomodel.

Autoreg	FPGA	iBiosim	speedup
Simulation time (sec.)	0.1	0.4	4x
End to End (including writing data from Ethernet to disk)	2.88	0.4	No speedup

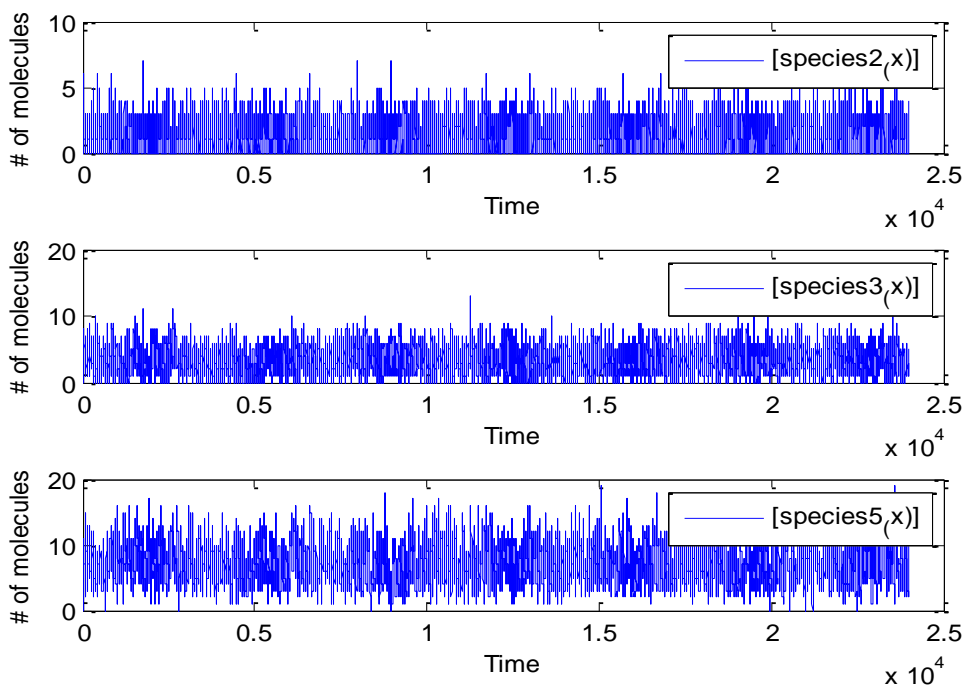
Στο παρακάτω μοντέλο δεν παρατηρήθηκε αύξηση στο speedup όταν προσμετρήθηκε και η καταγραφή των δεδομένων στο δίσκο. Αυτό συμβαίνει γιατί μιλάμε για ένα μοντέλο που σε 100 sec αντιδρά πλήρως άρα ο όγκος της πληροφορίας είναι μικρός και έχουμε λίγες αντιδράσεις μόνο 10 οπότε στο PC για αυτό το μοντέλο ο χρόνος είναι πιο γρήγορος από ότι στο δικό μας σύστημα που χρησιμοποιούμε για την μεταφορά των δεδομένων στην αργή θύρα Ethernet και δεχόμαστε και επιπλέον καθυστερήσεις για την εγγραφή των δεδομένων από την Ethernet στο δίσκο. Μέχρι στιγμής φαίνεται ως η μοναδική περίπτωση στην οποία δεν παρατηρείται speedup για μια end to end διαδικασία. Όμως υπολογίζουμε ότι χρησιμοποιώντας την Ethernet 1GB/sec θα καταφέρουμε στον ίδιο χρόνο ή και σε

Λιγότερο να έχουμε καλύτερα αποτελέσματα.

Επίσης παρατηρούμαι τις αντίστοιχες κυματομορφές για χρόνο 400min.



Σχήμα 75: Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων για χρόνο 400min.



Σχήμα 74: Αποτελέσματα προσομοίωσης Αυτοελεγχόμενων αναδραστικών Γονιδιακών Αντιδράσεων FRM4X SoC για χρόνο αντίδρασης 400 min συμπεριφορά στοιχείων 2x,3x,5x.

7.9 Αποτελέσματα και συμπεράσματα της διαδικασίας των πολλαπλών προσομοιώσεων.

Συγκρίνοντας τα αποτελέσματα του FRM4X SoC με τα ίδια μοντέλα από το Copassi και το Ibiosim και για τον ίδιο αριθμό προσομοιώσεων φαίνεται ότι αφενός συμπίπτουν τα αποτελέσματα μας και αφετέρου παρατηρείται αύξηση της απόδοσης σε χρόνο καθώς προκύπτει ότι σχεδόν σε όλες τις προσομοιώσεις το σύστημα ανταποκρίνεται γρηγορότερα. Αξίζει να σημειωθεί ότι τα προηγούμενα αποτελέσματα σημειώθηκαν με το αργό κομμάτι που δίδεται στην open core βιβλιοθήκη του Leon δηλαδή για 100 Mbit/sec εν αντιθέσει της πλήρης έκδοσης του 1GB/sec η οποία υπολογίζουμε ότι θα αύξανε περαιτέρω την απόδοση.

ΚΕΦΑΛΑΙΟ 8

ΕΠΙΛΟΓΟΣ

8. Γενικά συμπεράσματα και μελλοντικές βελτιώσεις

Η εργασία αυτή είχε ως στόχο τη δημιουργία μιας πλατφόρμας η οποία μπορεί να προσομοιώσει δίκτυα βιοχημικών μοντέλων χιλιάδων αντιδράσεων. Η ανάγκη αυτή όπως αποδείχθηκε έγκειται στο γεγονός ότι οι προσομοιωτές που υπάρχουν δεν μπορούν ακόμα να ανταπεξέλθουν πλήρως στη συνεχή ανάπτυξη και εξέλιξη του τομέα της συστημικής βιολογίας όπου παρατηρείται η τάση για ανάλυση πολύπλοκων συστημάτων βιοχημικών αντιδράσεων. Επίσης παραμετροποιώντας και παραλληλοποιώντας περαιτέρω τους αλγορίθμους στοχαστικής προσομοίωσης κατορθώσαμε να επιτύχουμε την μεγαλύτερη απόδοση σε σχέση πάντα με τα μέχρι τώρα στοιχεία της διεθνούς βιβλιογραφίας (2.6-3.1 Mega Reactions cycles/sec). Επιπλέον κατορθώσαμε και ενσωματώσαμε και το ερευνητικό πρόγραμμα SysPy με το οποίο ανάλογα με τις απαιτήσεις του μοντέλου, των απαιτήσεων του χρήστη και της διαθέσιμης τεχνολογίας μπορεί πολύ εύκολα να συνθέσει κυκλώματα προσομοιωτών κατάλληλα για την συγκεκριμένη εφαρμογή. Ακόμα όσο αναφορά την σχεδίαση καταφέραμε να παραμετροποιήσουμε την διαδικασία προσομοίωσης και έτσι για πρώτη φορά μπορούν να προσομοιωθούν μοντέλα βιοχημικά ανώτερα του 2^{ου} βαθμού(παραπάνω από 3 αντιδρώντα).

Συνοψίζοντας η έρευνα αυτή έδειξε ότι οι υλοποιήσεις σε υλικό (Hardware) δημιουργώντας πλατφόρμες ειδικού σκοπού κρίνονται αναγκαίες για τις σημερινές αλλά κυρίως για τις μελλοντικές ανάγκες της συστημικής βιολογίας. Ενδιαφέρον επίσης παρουσιάζει η αρχιτεκτονική FRM4X_NRM4X+Leon και εφόσον έγινε ένα πρώτο βήμα στην παραλληλοποίηση και παραμετροποίηση του NRM φαίνεται ότι θα μπορούσε να υπάρξει περαιτέρω έρευνα η οποία θα οδηγήσει σε θεαματική αύξηση της απόδοσης. Μιλώντας για απόδοση ορίζουμε σαν απόδοση τον αριθμό των αντιδράσεων που γίνονται ανά δευτερόλεπτο, το οποίο κρίνεται ιδιαίτερα σημαντικό στην συστημική βιολογία ώστε σε σύντομο χρονικό διάστημα να έχουμε καταγεγραμμένη όλο την ανάλυση της πορείας των βιοχημικών φαινομένων.

Επίσης λόγω της συνεχούς ανάπτυξης της τεχνολογίας των FPGAs που επιφέρει όλο και μεγαλύτερης χωρητικότητας FPGA σε αναλογία προγραμματιζόμενων πυλών αλλά και

αύξηση της συχνότητα λειτουργίας, φαίνεται ότι θα μπορούσαν να αναπτυχθούν περαιτέρω τεχνικές παραλληλοποιήσεις που αφορούν τους 2 αλγόριθμους FRM και NRM.

Κλείνοντας την εργασία αυτή φαίνεται ότι θα πρέπει να γίνει περαιτέρω μελέτη πάνω στον τομέα υλοποιήσεων προσομοιωτών στοχαστικής προσομοίωσης σε Hardware και πιο συγκεκριμένα πιστεύουμε ότι θα πρέπει να μελετηθεί καλύτερα το κομμάτι της αρχιτεκτονικής που υλοποιεί τον NRM αλγόριθμο καθώς φαίνεται ότι η απόδοση και το speedup αυξάνεται όσο προχωράμε σε μεγαλύτερα μοντέλα σε σχέση με αντίστοιχες εφαρμογές λογισμικού, καθώς επίσης ενδεχομένως να πρέπει να γίνει και μια πιο ενδελεχής μελέτη και σύγκριση των 2 αλγορίθμων FRM/NRM όσο αναφορά την προσομοίωση μεγάλων μοντέλων βιοχημικών αντιδράσεων. Επίσης η μελέτη του NRM αλγορίθμου θα πρέπει να επικεντρωθεί στις μεθόδους υπολογισμού του ελάχιστου χρόνου καθώς από αυτό το σημείο θα κριθεί η περαιτέρω επιτάχυνση του NRM για να ξεφύγουμε από την σταθερή επίδοση σε 1-Mega Reaction cycle/sec που αποδίδεται στην βιβλιογραφία.

Επιπλέον σαν μελλοντική προοπτική θεωρούμε αναγκαία την υλοποίηση Multiple Operation Network On chip στο οποίο θα μπορούν να ενσωματωθούν διάφοροι αλγόριθμοι στοχαστικής προσομοίωσης εκτός από τους NRM-FRM σε μία πλατφόρμα όπως ο Direct Method, ο Optimized Direct Method και ο wNRM. Ακόμα θεωρούμε ότι εκμεταλλευόμενοι τις ιδιότητες της στοχαστικής προσομοίωσης θα μπορούσαμε γρήγορα δημιουργώντας μία Software/Hardware πλατφόρμα να αναλύσουμε και να προσομοιώσουμε όλα τα φαινόμενα των βιολογικών αντιδράσεων όπως είναι φαινόμενα διάχυσης, κινήσεις-κατανομή μορίων στο πλέγμα, δυναμικό αντιδράσεων τα οποία θα μπορούν να χρησιμοποιηθούν για την ανάλυση πρωτεϊνικών δομών ή για την προσομοίωση βιολογικών νευρωνικών δικτύων.

Η συνεχής ανάπτυξη του τομέα της βιοπληροφορικής διέπει την εξέλιξη καινούργιων τεχνολογιών οι οποίες θα συνεισφέρουν στην μελέτη πολύπλοκων βιοχημικών συστημάτων για την ανάλυση ασθενειών, για την ανάλυση όλων των βιοχημικών μεταβολικών οδών που θα οδηγήσουν στην ανάπτυξη οργάνων και στην δημιουργία εξελιγμένων μορφών «έξυπνων» φαρμάκων.

ΛΙΣΤΑ ΑΚΡΩΝΥΜΙΩΝ

ASIC	Application Specific Integrated Circuit
BRAM	Block Random Access Memory
CME	Chemical Master Equation
CU	Control Unit
CPLD	Complex Programmable Logic Device
DDR2/3	Double Data Rate 2/3
DG	Dependency Graph
DM	Direct Method
FRM	First Reaction Method
FG	Flag Table
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
HDL	Hardware Description Language
JTAG	Joint Test Action Group
LUT	Look Up Table
MMU	Memory Management Unit
NRM	Next Reaction Method
PAL	Programmable Array Logic
PC	Personal Computer

PE	Processing Elements
PLA	Programmable Logic Array
PLD	Programmable Logic Device
PQ	Priority Queue
RAM	Random Access Memory
Regs	Registers
ROM	Read Only Memory
RT	Reaction Table
SDRAM	Synchronous Dynamic Random Access Memory
SoC	System on Chip
ST	Species Table
ST-block	Statistics Block
SysPy	System Python
SRAM	Static Random Access Memory
SSA	Stochastic Simulation Algorithm
UART	Universal Asynchronous Receiver/Transmitter

ΑΝΑΦΟΡΕΣ

- [1] D.T.Gillespie “Stochastic Simulation of Chemical Kinetics” Annu. Rev. Phys. Chem. Vol.58: 35-55, 2007
- [2] M. Gibson and J. Bruck, J. “Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels” Phys. Chem. A 104, 1876 ~2000
- [3] Logaras, E. Manolakos, E.S. SysPy: using Python for processor-centric SoC design Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference Dec. 2010 p 762 – 765.
- [4] Steven H. Kleinstein, Philip E Seiden, “Simulating the Immune System”, Computing in Science and Engineering, Vol. 2, Issue 4: July-August, 2000.
- [5] M.Yoshimi, Y.Osana, Y.Nishikawa, N.Iwanaga, A.Funahashi, N.H.Kitano, H.Amano, Y. Shibata, Y.Iwaoka, T.Kojima Japan “An FPGA implementation of high throughput stochastic simulator for large scale biochemical systems”, IEEE FPL pp. 1-6., 2006.
- [6] J. Keane, C. Bradley, C. Ebeling, “A Compiled Accelerator for Biological Cell Signaling Simulations” 12th ACM Int'l. Symp. on FPGAs, pp. 233-241, 2004.
- [7] H. Li, L. Petzold, “Efficient parallelization of the stochastic simulation algorithm for chemically reacting systems on the graphics processing unit”. Int J. High Performance Comp Appl vol 24(2):107, SAGE, May 2010.
- [8] C. Dittamo, D. Cangelosi, “Optimized parallel implementation of Gillespie’s first reaction method on graphics processing units” In Proc. of Int'l. Conf. on Computer Modeling and Simulation, pp. 156–61, 2009.
- [9] Guido Klingbeil, Radek Erban, Mike Giles and Philip K. Maini “STOCHSIMGPU: Parallel stochastic simulation for the Systems Biology Toolbox 2 for MATLAB” BIOINFORMATICS Vol. 00 no. 00 2010.
- [10] Hazapi Orsalia Georgia, Elias S. Manolakos « Scalable FRM-SSA SoC design for the simulation of Networks with thousands of Biochemical reactions in real time » FPL 2011 Chania Sept 5-7 Greece
- [11] COPASI Development Team, COPASI Documentation, <http://www.copasi.org/tiki-index.php>, (2006).
- [12] C. MYERS, N. BARKER, K. JONES, H. KUWAHARA , C. MADSEN, and N. NGUYEN, “iBioSim: a tool for the analysis and design of genetic circuits”, to appear in

Bioinformatics.<http://www.async.ece.utah.edu/people/index.html>.

[13] <http://sbml.org/Software/libSBML>

[14] <http://www.gaisler.com/cms>

[15] M.Yosimi, Y.Iwaoka, Y.Nishikawa, A.Funahashi, N.H. Kitano, “FPGA Implementation of a data-driven Stochastic Biochemical Simulator with the Next Reaction Method” IEEE FPL pp. 254-259, 2007. Biochemical Systems”, Proceedings of the 19th.IEEE(IDPS’05).

[16] A.J Lotka Undamed Oscillations Derived from the Law of mass action J Am Chem Soc (42) (1920),pp1595-1599.

[17].Alistair Rogers and Yves Gibon « Enzyme Kinetics: Theory and Practice»Department of Environmental Sciences, Brookhaven National Laboratory,Upton, NY USA J. Schwender (ed.), Plant Metabolic Networks, Springer Science+Business Media, LLC 2009

[18]Eric Guisbert, Christophe Herman, Chi Zen Lu, et al. “A chaperone network controls the heat shock response in E. coli” Genes Dev. 2004 18: 2812-2821

[19]Carole J Proctor, Douglas AGray “Explaining oscillations and variability in the p53-Mdm2 system”BMC Systems Biology 2008,2:75 18 August 2008

[20] Samik Ghosh, Preetam Ghosh, Kalyan Basu and Sajal K. Das “Modeling the stochastic dynamics of protein synthesis: A discrete event simulation approach” BIOT 2007

[21] In silico Modeling of Alpha-synuclein Oligomerization Effects on Neuronal Homeostasi E. Ouzounoglou, K. Vekrellis, E. Manolakos Abstract Book p. 291, PS 578

[22] Carol e J. Proctor, Paul J. Tangem , Helen C. Ardley “Modelling the Role of UCH-L1 on Protein Aggregation in Age-R elated Neurodegeneration”.

[23] A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. J. Computational Physics 22:403-434 (1976).

[24]Ofer Malcai, Ofer Biham, Peter Richmond, and Sorin Solomon,Racah Institute of Physics, The Hebrew University, Jerusalem 91904,Israel Department of Physics, Trinity College, Dublin, Ireland «Theoretical Analysis and Simulations of the Generalized Lotka-Volterra Model» arXiv:cond-mat/0208514v1 [cond-mat.stat-mech] 27 Aug 2002 Cornell University Library.

[25] Hiroyuki Kurata Hana El-Samad ,Rei Iwasaki¹, Hisao Ohtake, John C. Doyle, Irina Grigorova, Carol A. Gross, Mustafa Khammash “Module-Based Analysis of Robustness Tradeoffs in the Heat Shock Response System” PLoS Computational Biology July 2006

- [26] Thomas W Evans.,Colin S. Gillespie, Darren J. Wilkinson. “The SBML Discrete Stochastic Models Test Suite” Oxford University Press, Vol 0 no 00. pp1-2, 2007.
- [27] Yang Cao, Hong Li , Linda Petzold,“Efficient formulation of the stochastic simulation algorithm for chemically reacting systems” J. Chem. Phys., Vol. 121, No. 9, 1 September 2004.
- [28]Xianning Liu, Lansun Chen “Complex dynamics of Holling type II Lotka–Volterra predator–prey system with impulsive perturbations on the predator”Chaos,Solitons and Fractals Volume 16, Issue 2, March 2003, Pages 311-320
- [29] Carole J Proctor, Douglas AGray “Explaining oscillations and variability in the p53-Mdm2 system”BMC Systems Biology 2008,2:75 18 August 2008
- [30]Dr.Mohammed M. Alani,“ Testing Randomness in Ciphertext of Block-Ciphers Using DieHard Tests”, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.4, April 2010.
- [31]Yang Kuang, Edoardo Beretta, “Global qualitative analysis of a ratio-dependent predator–prey system”. Department of Mathematics, Arizona State University, Tempe, AZ 85287, USA Istituto di Biomatemica, Universita di Urbino, I-61029 Urbino, Italy J. Math. Biol. (1998) 36: 389D406
- [32]A Markov Contingency able model for replicated Lotka-Voltera Systems near equilibrium Joel E. Society of Fellows, Harvard University, Cambridge, Massachusetts 02138 Vol. 104, No. 940 November-December 1970
- [33].Alistair Rogers and Yves Gibon “Enzyme Kinetics: Theory and Practice” Department of Environmental Sciences, Brookhaven National Laboratory,Upton, NY USA J. Schwender (ed.), Plant Metabolic Networks, Springer Science+Business Media, LLC 2009
- [34] Y. Cao, H. Li, and L. Petzold. “Efficient formulation o f the stochastic simulation a lgorithm for chemically reacting systems”,JChemPhys,121:4059–4067, 2004.
- [35].Yang Kuang, Edoardo Beretta, “Global qualitative analysis of a ratio-dependent predator–prey system”. Department of Mathematics, Arizona State University, Tempe, AZ 85287, USA Istituto di Biomatemica, Universita di Urbino, I-61029 Urbino, Italy J. Math. Biol. (1998) 36: 389D406
- [36] Joel E. Cohen “A Markov consistency-table model for replicated Lotka-Voltera systems near equilibrium” Society of Fellows, Harvard University, Cambridge, Massachusetts 02138 Vol. 104, No. 940 November-December 1970

- [37] Carole J Proctor, Douglas A Gray “Explaining oscillations and variability in the p53-Mdm2 system” BMC Systems Biology 2008,2:75 18 August 2008
- [38] Y. Cao, H. Li, and L. Petzold. “Efficient formulation of the stochastic simulation algorithm for chemically reacting systems”, JChemPhys, 121:4059–4067, 2004.
- [39] H. Kurata, H. El-Samad, T.-M. Yi, M. Khammash, and J. Doyle. “Feedback regulation of the heat shock response in E. coli”, Decision and Control, 2001. Proceedings of the 40th IEEE Conference, 1:837–842, 2001.
- [40] E. J. Crampin and S. Schnell, “New approaches to modelling and analysis of biochemical reactions, pathways and networks”, Progress in Biophysics & Molecular Biology, 86 (2004), pp. 1-4.
- [41] C. M. Evans^{a,b} and G. L. Findley “Analytic solutions to the Lotka-Volterra model for sustained chemical oscillation” Center for Advanced Microstructures and Devices (CAMD) and Department of Chemistry, Louisiana State University, Baton Rouge, LA 70803, Department of Chemistry, University of Louisiana at Monroe, Monroe, LA 71209
- [42] Debabrata Dutta and J K Bhattacharjee “Lotka-Volterra dynamics under periodic influence”
- [43] Xu and Cai: “Weighted next reaction method and parameter selection for efficient simulation of rare events in biochemical reaction systems “. EURASIP Journal on Bioinformatics and Systems Biology 2011 2011 :4
- [44] Mettetal J.T, D. Muzzey, J. M. Pedraza, E. M. Ozbudak, and A. van Oudenaarden. “Predicting stochastic gene expression dynamics in single cells”. PNAS 103(19):7304-9, 2006.
- [45] W. E, D. Liu, and E. Vanden-Eijnden, “Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales”, J. Chem. Phys., 123 (2005), p. 194107
- [46] Chris J Meyers Engineering Genetic circuits Chapman & Hall CRC Mathematical and Computational Biology
- [47] Wang Xiang-Sun, Luonan Chen, Rui-Sheng, “Biomolecular Networks Methods and Applications in System’s Biology”