

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗ ΜΙΚΡΟΗΛΕΚΤΡΟΝΙΚΗ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση πρωτόκολλου επικοινωνίας SPI σε τεχνολογία CMOS 0,15 μm με σχεδίαση κυκλωμάτων ανθεκτικών σε ιονίζουσες ακτινοβολίες

Ιωάννης Ε. Κορμανός

Επιβλέποντες: Αραπογιάννη Αγγελική, Καθηγητής Λαμπρόπουλος Χάρης, Καθηγητής

AOHNA

ΦΕΒΡΟΥΑΡΙΟΣ 2019

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Υλοποίηση πρωτόκολλου επικοινωνίας SPI σε τεχνολογία CMOS 0,15 μm με σχεδίαση κυκλωμάτων ανθεκτικών σε ιονίζουσες ακτινοβολίες

Ιωάννης Ε. Κορμανός

A.M.: MM247

ΕΠΙΒΛΕΠΟΝΤΕΣ: Αραπογιάννη Αγγελική, Καθηγητής Λαμπρόπουλος Χάρης, Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: Αραπογιάννη Αγγελική, Καθηγητής Λαμπρόπουλος Χάρης, Καθηγητής

Φεβρουάριος 2019

ΠΕΡΙΛΗΨΗ

Οι επιπτώσεις της ιονίζουσας ακτινοβολίας στα μικροηλεκτρονικά κυκλώματα είναι σημαντικό ζήτημα για την αξιοπιστία τους στις διαστημικές εφαρμογές.

Το αντικείμενο της παρούσας μεταπτυχιακής διπλωματικής εργασίας, είναι η υλοποίηση σειριακού πρωτόκολλου SPI (Serial Peripheral Interface) με υποτελή (slave) τρόπο λειτουργίας, με χρήση κυκλωματικών τεχνικών που να εξασφαλίζουν ανθεκτικότητα στα φαινόμενα απλού συμβάντος (Single Event Effects - SEE), τα οποία προκαλούνται από τα φορτισμένα σωματίδια της κοσμικής ακτινοβολίας στο διαστημικό περιβάλλον. Αυτή η ανθεκτικότητα θα εξασφαλισθεί αφενός με τροποποίηση της φυσικής σχεδίασης (layout) των προτύπων κελιών (standard cells) που παρέχει το εργοστάσιο κατασκευής των CMOS κυκλωμάτων και αφετέρου με χρήση τριπλού πλεονασμού (Triple Mode Redundancy - TMR) στην αρχιτεκτονική του κυκλώματος. Η τροποποίηση των προτύπων κελιών αφορούσε την πρόσθεση δακτυλίων προστασίας (guard rings) έτσι ώστε να μειωθεί η πιθανότητα ανοικτοκλειδώματος (latchup).

Τα τροποποιημένα πρότυπα κελιά χαρακτηρίστηκαν πλήρως σε επίπεδο προσομοίωσης. Με τα αποτελέσματα του χαρακτηρισμού παράχθηκαν τα .lib αρχεία που είναι απαραίτητα για την προσομοίωση με την χρήση ψηφιακών προσομοιωτών.

Χρησιμοποιώντας τη μέθοδο TMR, έγιναν δύο υλοποιήσεις του SPI slave με τη χρήση γλώσσας περιγραφής υλικού VHDL. Στη συνέχεια προχωρήσαμε στη λογική σύνθεση των κυκλωμάτων με τη χρήση των τροποποιημένων προτύπων κελιών και στην φυσική σχεδίαση (layout) των υλοποιήσεων, ώστε να έχουμε τα τελικά αποτελέσματα και να γίνουν οι απαραίτητες συγκρίσεις.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ολοκληρωμένα κυκλώματα ανθεκτικά σε ιονίζουσα ακτινοβολία

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Δακτύλιοι προστασίας, Τριπλός πλεονασμός υποσυστημάτων, Σειριακό πρωτόκολλο επικοινωνίας SPI, Χαρακτηρισμός πρότυπων κελιών, Φαινόμενα απλού συμβάντος

ABSTRACT

Radiation effects on microelectronics are an important reliability issue for space applications.

The subject of this thesis is the implementation of the slave of the Serial Peripheral Interface (SPI) protocol, with the appropriate circuit design in order to guarantee hardness for the Singe Event Effects (SEE), caused by charged particles of cosmic radiation in the space environment. This hardness can be ensured on the one hand by modification of the layout of the foundry standard cells, with the introduction of guard rings and on the other hand by the usage of Triple Mode Redundancy (TMR) in the circuit architecture. The addition of guard rings reduces the possibility of latch-up.

The modified standard cells were fully characterized at simulation level. With the results of this characterization the .lib files were produced. These files are necessary for the simulation with the aid of digital simulators.

Using the hardware description language VHDL, two TMR implementations of the SPI slave block were designed. The two implementations were synthesized at gate level and their layout was generated using the modified standard cells in order to compare the performance of the final circuits.

SUBJECT AREA: Radiation hard integrated circuits

KEYWORDS: Guard rings, Singe Event Effects (SEE), Triple Mode redundancy (TMR), SPI protocol, Standard cell characterization

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις ευχαριστίες μου στην καθηγήτρια κα Αγγελική Αραπογιάννη, αλλά και τους υπόλοιπους καθηγητές του μεταπτυχιακού για τις γνώσεις που μου προσέφεραν. Επίσης θα ήθελα να ευχαριστήσω τον Γιάννη Γλυκιώτη από την εταιρία ADVEOS για την πολύτιμη βοήθεια του που χρειάστηκα αρκετές φορές και τον Αλέξανδρο Παπαγγελή που έφτιαξε τα θεμέλια ώστε να μπορέσει να γίνει αυτή η εργασία.

Θα ήθελα να ευχαριστήσω ιδιαιτέρως τον καθηγητή κ. Χάρη Λαμπρόπουλο που μου έδωσε την ευκαιρία μέσω αυτής της εργασίας να επεκτείνω τις γνώσεις μου και για την καθοδήγηση του καθ' όλη τη διάρκεια της εργασίας.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου και την αγαπημένη μου Αμέρισσα για την συμπαράσταση τους σε αυτό το μακρύ και δύσκολο ταξίδι.

ΠΕΡΙΕΧΟΜΕΝΑ

ПР	ΠΡΟΛΟΓΟΣ		
1.	ΕIΣ	ΑΙ ΩΙ Η	23
1.1	Περι	βάλλον διαστημικής ακτινοβολίας	23
1	1.1	Ζώνες ακτινοβολίας Βαν Άλεν	23
1	.1.2	Ηλιακές εκλάμψεις	23
1	.1.3	Ηλιακός άνεμος	24
1	.1.4	Κοσμικές ακτίνες	24
1.2	Επιδ	ράσεις απλού συμβάντος στις ηλεκτρονικές συσκευές και συστήματα	24
1	.2.1	Προσδιορισμός επιδράσεων απλού συμβάντος	25
1	.2.2	Κλείδωμα απλού συμβάντος (SEL)	26
1	.2.3	Μη καταστροφικές επιδράσεις	28
	1.2.3	.1 Διατάραξη απλού συμβάντος (SEU)	29
	1.2.3	.2 Μετάβαση απλού συμβάντος (SET)	30
1.3	Επιζ	ράσεις ολικής δόσης ιονισμού (Total Ionizing dose - TID)	30
1	.3.1	Αύξηση των παρασιτικών ρευμάτων	32
1.4	Ανθε	κτικότητα σε επιδράσεις ολικής δόσης	33
1.5	Ανθε	κτικότητα σε επιδράσεις απλού συμβάντος	35
1	.5.1	Ανθεκτικότητα σε διαταράξεις απλού συμβάντος (SEU)	35
1	.5.2	Ανθεκτικότητα σε κλείδωμα απλού συμβάντος (SEL)	36
2.	XAF	ΡΑΚΤΗΡΙΣΜΟΣ ΤΩΝ ΚΕΛΙΩΝ	38
2.1	Εισα	γωγή	
2.2	Ακρ	ότατα σημεία διεργασίας κατασκευής ολοκληρωμένων κυκλωμάτων	38
2.3	Ροή	χαρακτηρισμού	39
2.4	Mov	τελοποίηση [39]	42
2	.4.1	Μοντέλο χρονισμού	43
2	.4.2	Χρονικά τόξα (Timing arcs)	43
	2.4.2	.1 Τόξα καθυστέρησης	44
	2.4.2	.2 Συνδυαστικά τόξα καθυστέρησης	44
	2.4.2	.3 Ακολουθιακά τόξα καθυστέρησης	45

2.	4.3	Τόξα	χρονικού περιορισμού (constraint arcs)	46
	2.4.3	.1	Ακολουθιακά τόξα χρονικού περιορισμού	46
	2.4.3	.2	Μη ακολουθιακά τόξα χρονικού περιορισμού	46
	2.4.3	.3	Τόξα χρονικού περιορισμού στο μοντέλο NLDM	46
2.	4.4	Моут	έλο ισχύος	47
	2.4.4	.1	Στατική ισχύς	48
	2.4.4	.2	Στατική ισχύς κελιού	48
	2.4.4	.3	Δυναμική ισχύς	48
	2.4.4	.4	Ισχύς μεταγωγής (switching power)	48
	2.4.4	.5	Κρυφή ισχύς (Hidden Power)	48
2.5	Κελι	ά ποι	υ θα χαρακτηριστούν	49
2.	5.1	Συνδ	υαστικά κελιά	50
2.	5.2	Ακολ	ουθιακά κελιά	53
2.6	Χαρ	ακτηρ	ισμός	55
2.	6.1	Δoµŕ	του σχεδιότυπου (template)	55
2.7	Απο	τελέσ	ματα χαρακτηρισμού	56
2.	7.1	Αποι	ελέσματα για το κελί AND2_X0d5	56
2.	7.2	Αποι	ελέσματα για το κελί NOR2_X0d5	59
2.	7.3	Αποι	ελέσματα για το κελί XOR2_X1	61
2.	7.4	Αποι	ελέσματα για το κελί INV_X1	64
2.	7.5	Αποι	ελέσματα για το κελί CLKBUF_X1	66
2.	7.6	Αποι	ελέσματα για το κελί ΗΑ_Χ1	69
2.	7.7	Αποι	ελέσματα για το κελί ΑΟ21_Χ0d5	72
2.	7.8	Αποι	ελέσματα για το κελί AOI21_X0d5	74
2.	7.9	Αποι	ελέσματα για το κελί OAl22_X0d5	77
2.	7.10	A٦	τοτελέσματα για το κελί DFC_X1	79
2.	7.11	A٦	τοτελέσματα για το κελί DFNC_X1	83
2.	7.12	٦A	τοτελέσματα για το κελί DFCS_X1	86
3.	ΥΛΟ	опоі	ΗΣΗ ΤΟΥ SPI SLAVE	91
3.1	Ποω	τόκο	λλο SPI [40]	Q1
0.1	Πpω	TORO		
3.2	Λειτα	ουργί	α τριπλού πλεονασμού (TMR) [42]	92
3.3	Υλοι	τοίηο	η του SPI slave με χρήση τριπλού πλεονασμού	93
3.	3.1	Προδ	ιαγραφές	94
3.	3.2	Σύνθ	εση του slave	95
	3.3.2	.1	Ψηφοφόρος (Voter)	95
	3.3.2	.2	Καταχωρητές με ψηφοφόρο	96
	3.3.2	.3	Μετρητής των bit διεύθυνσης	97

	3.3.2.4	Μετρητής των bit δεδομένων	99
	3.3.2.5	Καταχωρητές ολίσθησης	100
	3.3.2.6	Κεντρική μονάδα επεξεργασίας	102
	3.3.2.7	Ανώτερο επίπεδο του slave (Top level)	107
3	.3.3 Φυσ	κή σχεδίαση (layout) του slave	108
	3.3.3.1	Φυσική σχεδίαση του slave με χρήση απλού TMR	108
	3.3.3.2	Φυσική σχεδίαση (layout) του slave με χρήση πλήρους TMR	110
3.4	Προσομο	ιώσεις	112
3.5	Αποτελέα	ψατα σύνθεσης και φυσικής σχεδίασης	115
3	.5.1 Απο	τελέσματα για την υλοποίηση με χρήση απλού TMR	115
3	.5.2 Απο ⁻	τελέσματα για την υλοποίηση με χρήση πλήρους TMR	119
4.	ΣΥΝΟΨ	Η ΑΠΟΤΕΛΕΣΜΑΤΩΝ – ΣΥΜΠΕΡΑΣΜΑΤΑ	123
4.1	Συνοπτικ	ά αποτελέσματα χαρακτηρισμού	123
4.2	Συνοπτικ	ά αποτελέσματα της υλοποίησης του slave	128
4.3	Συμπερά	σματα	128
ПІМ	ΙΑΚΑΣ ΟΙ	ΡΟΛΟΓΙΑΣ	130
ΣΥΙ	ΝΤΜΗΣΕΙ	Σ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	131
ПА	PAPTHM	ΑΙ	132
ПА	PAPTHM	A II	137
			172

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1.1: Βασικός μηχανισμός SEL σε έναν αντιστροφέα CMOS. [3]26
Σχήμα 1.2: Ισοδύναμη διάταξη της ΡΝΡΝ παρασιτικής δομής. [3]
Σχήμα 1.3: Ενεργοποίηση PNPN δομής. Το ιόν διασχίζει την επαφή Ν-Πηγαδιού / Ρ- υποστρώματος. Το ρεύμα ενισχύθηκε στη δομή θετικής ανάδρασης. [3]
Σχήμα 1.4: Καμπύλη ρεύματος - τάσης για την διάταξη pnpn. [6]
Σχήμα 1.5: Σχηματισμός ζευγών ηλεκτρονίων οπών κατά την πρόσκρουση του σωματιδίου πάνω στο υλικό. [6]28
Σχήμα 1.6: Σωματίδιο προσκρούει πάνω στον απαγωγό ενός τρανζίστορ και παράγει μια αιχμή στο ρεύμα εξόδου. [6]29
Σχήμα 1.7: Απλουστευμένο διάγραμμα μιας μονάδας μνήμης. [6]
Σχήμα 1.8: Σχηματικό διάγραμμα ενός MOSFET τύπου-n σε κανονική λειτουργία. [6]31
Σχήμα 1.9: Σχηματικό διάγραμμα ενός MOSFET τύπου-η μετά την ακτινοβολία. [6]31
Σχήμα 1.10: Κάτοψη ενός τρανζίστορ όπου φαίνονται και τα παρασιτικά ρεύματα. [6] .32
Σχήμα 1.11: Δημιουργία των παρασιτικών ρευμάτων από την πηγή στον απαγωγό. [6]
Σχήμα 1.12: Επίδραση της συνολικής δόσης στην τάση κατωφλίου ενός τρανζίστορ συναρτήσει του πάχους οξειδίου. [6]33
Σχήμα 1.13: Κάτοψη της κανονικής και της κυκλικής γεωμετρίας. [6]
Σχήμα 1.14: Λεπτομέρεια στην γεωμετρία ενός κυκλικού τρανζίστορ. [6]
Σχήμα 1.15: Σχεδίαση τρανζίστορς με χρήση δακτυλίων προστασίας. [6]
Σχήμα 1.16: Αύξηση ανθεκτικότητας σε SEU με πλεονασμό: Το TMR εφαρμόζεται σε όλες τις πληροφορίες καθώς και στους ψηφοφόρους. [26]
Σχήμα 1.17: CMOS τρανζίστορς με δακτύλιο προστασίας. [32]
Σχήμα 2.1: Ακρότατα σημεία. [34]39
Σχήμα 2.2: Διάγραμμα ροής χαρακτηρισμού. [35]39
Σχήμα 2.3: Χρόνος μετάβασης ενός σήματος. [37]41
Σχήμα 2.4: Καθυστέρηση διάδοσης μίας πύλης AND. [38]41
Σχήμα 2.5: Κατανόηση των χρόνων προετοιμασίας και συγκράτησης. [37]41

Σχήμα 2.6: Χρόνοι αποκατάστασης και απομάκρυνσης ενός flip-flop. [37]42
Σχήμα 2.7: Χρονικά τόξα σε μία πύλη AND. [39]43
Σχήμα 2.8: NLDM τόξα καθυστέρησης44
Σχήμα 2.9: Τόξα χρονικού περιορισμού47
Σχήμα 2.10: Τόξα δυναμικής ισχύος49
Σχήμα 2.11: Τροποποιημένα layout των κελιών με προσθήκη δακτυλίων προστασίας. Α) AND2_X0d5, B) NOR2_X0d5, Γ) XOR2_X1, Δ) INV_X1 και Δ) CLKBUF_X150
Σχήμα 2.12: Σχηματικό κελιού AO21_X0d551
Σχήμα 2.13: Σχηματικό κελιού OAI22_X0d551
Σχήμα 2.14: Σχηματικό κελιού AOI21_X0d551
Σχήμα 2.15: Τροποποιημένα layout των κελιών με προσθήκη δακτυλίων προστασίας. Α) AO21_X0d5, B) OAI22_X0d5 και Γ) AOI21_X0d552
Σχήμα 2.16: Τροποποιημένο layout του κελιού ΗΑ_Χ1 με προσθήκη δακτυλίων προστασίας
Σχήμα 2.17: Τροποποιημένο layout του κελιού DFC_X1 με προσθήκη δακτυλίων προστασίας53
Σχήμα 2.18: Τροποποιημένο layout του κελιού DFNC_X1 με προσθήκη δακτυλίων προστασίας54
Σχήμα 2.19: Τροποποιημένο layout του κελιού DFCS_X1 με προσθήκη δακτυλίων προστασίας54
Σχήμα 2.20: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d556
Σχήμα 2.21: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί AND2_X0d5.
Σχήμα 2.22: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d5
Σχημα 2.23: Συγκριση καθυστερησης μεταξυ σημείων FF και SS για το κελί AND2_X0d5.
Σχήμα 2.24: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d557

Σχήμα 2.25: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί AND2_X0d5......58 Σχήμα 2.26: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d5......58 Σχήμα 2.27: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί AND2_X0d5......58 Σχήμα 2.28: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5......59 Σχήμα 2.29: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί Σχήμα 2.30: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5......59 Σχήμα 2.31: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί NOR2 X0d5. Σχήμα 2.32: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5.....60 Σχήμα 2.33: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί NOR2_X0d5......60 Σχήμα 2.34: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5.....61 Σχήμα 2.35: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί NOR2_X0d5.....61 Σχήμα 2.36: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X1.....61 Σχήμα 2.37: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί XOR2_X1. Σχήμα 2.38: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X1.....62 Σχήμα 2.39: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί XOR2 X1.

Σχήμα 2.40: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X163
Σχήμα 2.41: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί XOR2_X163
Σχήμα 2.42: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X163
Σχήμα 2.43: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί XOR2_X164
Σχήμα 2.44: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X164
Σχήμα 2.45: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί INV_X1.64
Σχήμα 2.46: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί INV_X1…64
Σχήμα 2.47: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X165
Σχήμα 2.48: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X165
Σχήμα 2.49: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί INV_X165
Σχήμα 2.50: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X166
Σχήμα 2.51: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί INV_X166
Σχήμα 2.52: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X166
Σχήμα 2.53: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί CLKBUF_X167
Σχήμα 2.54: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X167
Σχήμα 2.55: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί CLKBUF_X1.

Σχήμα 2.56: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X168
Σχήμα 2.57: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί CLKBUF_X168
Σχήμα 2.58: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί CLKBUF_X168
Σχήμα 2.59: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X169
Σχήμα 2.60: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ169
Σχήμα 2.61: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί HA_X169
Σχήμα 2.62: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ170
Σχήμα 2.63: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί HA_X170
Σχήμα 2.64: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ170
Σχήμα 2.65: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί HA_X171
Σχήμα 2.66: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ171
Σχήμα 2.67: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί HA_X171
Σχήμα 2.68: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AO21_X0d572
Σχήμα 2.69: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί AO21_X0d5. 72
Σχήμα 2.70: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΑΟ21_Χ0d572
Σχήμα 2.71: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί AO21_X0d5.

Σχήμα 2.72: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AO21_X0d573
Σχήμα 2.73: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί AO21_X0d573
Σχήμα 2.74: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AO21_X0d574
Σχήμα 2.75: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί AO21_X0d574
Σχήμα 2.76: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AOI21_X0d574
Σχήμα 2.77: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί AOI21_X0d575
Σχήμα 2.78: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΑΟΙ21_Χ0d575
Σχήμα 2.79: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί AOI21_X0d5.
Σχήμα 2.80: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΑΟΙ21_Χ0d576
Σχήμα 2.81: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί AOI21_X0d5
Σχήμα 2.82: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AOI21_X0d576
Σχήμα 2.83: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί AOI21_X0d5
Σχήμα 2.84: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί OAI22_X0d5
Σχήμα 2.85: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί OAl22_X0d5
Σχήμα 2.86: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί OAI22_X0d5.

Σχήμα 2.87: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΟΑΙ22_Χ0d5
Σχήμα 2.88: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΟΑΙ22_Χ0d5
Σχήμα 2.89: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί ΟΑΙ22_X0d5
Σχήμα 2.90: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΟΑΙ22_Χ0d5
Σχήμα 2.91: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί ΟΑΙ22_X0d5
Σχήμα 2.92: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X180
Σχήμα 2.93: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί DFC_X1.
Σχήμα 2.94: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X180
Σχήμα 2.95: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί DFC_X1. 80
Σχήμα 2.96: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X181
Σχήμα 2.97: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί DFC_X181
Σχήμα 2.98: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X182
Σχήμα 2.99: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί DFC_X182
Σχήμα 2.100: Διάγραμμα σύγκρισης χρονικών περιορισμών μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X182
Σχήμα 2.101: Σύγκριση χρονικών περιορισμών μεταξύ σημείων FF και SS για το κελί DFC_X183
Σχήμα 2.102: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X183

Σχήμα 2.103: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί DFNC_X1. Σχήμα 2.104: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1......84 Σχήμα 2.105: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί DFNC X1. Σχήμα 2.106: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1......84 Σχήμα 2.107: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί DFNC_X1......85 Σχήμα 2.108: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1......85 Σχήμα 2.109: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί DFNC_X1......85 Σχήμα 2.110: Διάγραμμα σύγκρισης χρονικών περιορισμών μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1......86 Σχήμα 2.111: Σύγκριση χρονικών περιορισμών μεταξύ σημείων FF και SS για το κελί Σχήμα 2.112: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1.....87 Σχήμα 2.113: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί DFCS X1. Σχήμα 2.114: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1......87 Σχήμα 2.115: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί DFCS_X1. Σχήμα 2.116: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1......88 Σχήμα 2.117: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί

Σχήμα 2.118: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X189
Σχήμα 2.119: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί DFCS_X189
Σχήμα 2.120: Διάγραμμα σύγκρισης χρονικών περιορισμών μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X189
Σχήμα 2.121: Σύγκριση χρονικών περιορισμών μεταξύ σημείων FF και SS για το κελί DFCS_X190
Σχήμα 3.1: SPI διάταξη με τρείς slaves. [41]91
Σχήμα 3.2: Διάγραμμα χρονισμών με τους τρόπους λειτουργίας του SPI. [40]92
Σχήμα 3.3: Λειτουργία τριπλού πλεονασμού με ψηφοφορία. [42]92
Σχήμα 3.4: Τριπλός πλεονασμός υποσυστημάτων. [42]93
Σχήμα 3.5: Υποσύστημα με χρήση τριπλού πλεονασμού σε επίπεδο πύλης. [42]93
Σχήμα 3.6: Λογικό κύκλωμα ψηφοφόρου95
Σχήμα 3.7: Σχηματικό ψηφοφόρου μετά την σύνθεση96
Σχήμα 3.8: Σχηματικό καταχωρητή με χρήση απλού TMR96
Σχήμα 3.9: Σχηματικό καταχωρητή με χρήση πλήρους TMR97
Σχήμα 3.10: Σχηματικό μετρητή κύκλων ρολογιού για την αποστολή διεύθυνσης με χρήση απλού TMR97
Σχήμα 3.11: Σχηματικό μετρητή κύκλων ρολογιού για την αποστολή διεύθυνσης με χρήση πλήρους TMR98
Σχήμα 3.12: Σχηματικό μετρητή κύκλων ρολογιού για την μεταφορά δεδομένων με χρήση απλού TMR
Σχήμα 3.13: Σχηματικό μετρητή κύκλων ρολογιού για την μεταφορά δεδομένων με χρήση πλήρους TMR100
Σχήμα 3.14: Σχηματικό καταχωρητή ολίσθησης διευθύνσεων
Σχήμα 3.15: Σχηματικό καταχωρητή ολίσθησης με χρήση απλού TMR101
Σχήμα 3.16: Σχηματικό καταχωρητή ολίσθησης με χρήση πλήρους TMR102
Σχήμα 3.17: Σχηματικό καταχωρητή ολίσθησης δεδομένων

Σχήμα 3.18: Διάγραμμα καταστάσεων FSM για την αποθήκευση της διεύθυνσης104
Σχήμα 3.19: Διάγραμμα καταστάσεων FSM για την εγγραφή δεδομένων105
Σχήμα 3.20: Διάγραμμα καταστάσεων FSM για την ανάγνωση δεδομένων106
Σχήμα 3.21: Σχηματικό κεντρικής μονάδας επεξεργασίας του slave107
Σχήμα 3.22: Σχηματικό SPI slave με χρήση απλού TMR107
Σχήμα 3.23: Σχηματικό SPI slave με χρήση πλήρους TMR108
Σχήμα 3.24: Φυσική σχεδίαση του slave σε υλοποίηση με χρήση απλού TMR109
Σχήμα 3.25: Δέντρο ρολογιού του slave για υλοποίηση με χρήση απλού TMR109
Σχήμα 3.26: Κατανομή κελιών και μονάδων του slave για υλοποίηση με χρήση απλού TMR110
Σχήμα 3.27: Φυσική σχεδίαση του slave σε υλοποίηση με χρήση πλήρους TMR111
Σχήμα 3.28: Δέντρο ρολογιού του slave για υλοποίηση με χρήση πλήρους TMR111
Σχήμα 3.29: Κατανομή κελιών και μονάδων του slave για υλοποίηση με χρήση πλήρους TMR112
Σχήμα 3.30: Κυματομορφές καταχωρητών με ψηφοφόρους113
Σχήμα 3.30: Κυματομορφές καταχωρητών με ψηφοφόρους

Σχήμα 4.6: Γράφημα για τη σύγκριση του χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης......125

Σχήμα 4.7: Γράφημα για τη σύγκριση του χρόνου μετάβασης εξόδου της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS......125

Σχήμα 4.9: Γράφημα για τη σύγκριση της στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης......126

Σχήμα 4.10: Γράφημα για τη σύγκριση της στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης......126

Σχήμα 4.11: Γράφημα για τη σύγκριση της στατικής ισχύος της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS......126

Σχήμα 4.12: Στατιστικά για τη σύγκριση της στατικής ισχύος της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS......127

Σχήμα 4.15: Στατιστικά για τη σύγκριση της κατανάλωσης ισχύος της τροποποιημέ	ένης
βιβλιοθήκης για ακρότατα σημεία FF και SS	127
Σχήμα 4.16: Γράφημα για τη σύγκριση της κατανάλωσης ισχύος της τροποποιημέ	ένης
βιβλιοθήκης για ακρότατα σημεία FF και SS	128

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Πίνακας αληθείας ψηφοφόρου95
Πίνακας 2: Πίνακας αληθείας συνδυαστικού κυκλώματος του μετρητή των bit διεύθυνσης.
Πίνακας 3: Πίνακας αληθείας συνδυαστικού κυκλώματος του μετρητή των bit δεδομένων.
Πίνακας 4: Πίνακας καταστάσεων FSM για την αποθήκευση διεύθυνσης105
Πίνακας 5: Πίνακας καταστάσεων FSM για την εγγραφή δεδομένων στη μνήμη106
Πίνακας 6: Πίνακας καταστάσεων FSM για την ανάγνωση δεδομένων από τη μνήμη.106
Πίνακας 7: Λειτουργία ψηφοφόρου113
Πίνακας 8: Στατιστικά κατανάλωσης ισχύος σε απλό TMR για υλοποίηση με τροποποιημένα και πρωτότυπα κελιά119
Πίνακας 9: Στατιστικά κατανάλωσης ισχύος σε απλό TMR για ακρότατα σημεία FF και SS119
Πίνακας 10: Στατιστικά κατανάλωσης ισχύος σε πλήρες TMR για υλοποίηση με τροποποιημένα και πρωτότυπα κελιά122
Πίνακας 11: Στατιστικά κατανάλωσης ισχύος σε πλήρες TMR για ακρότατα σημεία FF και SS122
Πίνακας 12: Συνοπτικά αποτελέσματα της υλοποίησης του slave128

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία πραγματοποιήθηκε στο εργαστήριο αισθητήρων ηλεκτρονικών και τηλεπικοινωνιών του τμήματος Μηχανικών τεχνολογίας αεροσκαφών του ΤΕΙ Στερεάς Ελλάδας και στην εταιρεία "ADVEOS μικροηλεκτρονικά συστήματα" κατά τη διάρκεια της φοίτησης στο Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών στη Μικροηλεκτρονική, με συμμετέχοντες φορείς το Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών (Τμήματα Πληροφορικής και Τηλεπικοινωνιών, και Φυσικής), το Εθνικό Μετσόβιο Πολυτεχνείο (Ερευνητικό Πανεπιστημιακό Ινστιτούτο Συστημάτων Επικοινωνιών και Υπολογιστών), το Εθνικό Κέντρο Έρευνας Φυσικών Επιστημών Δημόκριτος (Ινστιτούτο Νανοεπιστήμης και Νανοτεχνολογίας) και το Εθνικό Πολυτεχνικό Ινστιτούτο της Γκρενόμπλ.

Στο πρώτο κεφάλαιο θα αναφερθούμε στα φαινόμενα που μπορεί να προκληθούν από τη διαστημική ακτινοβολία. Συγκεκριμένα θα δούμε τις επιδράσεις απλού συμβάντος και τις επιδράσεις ολικής δόσης, καθώς και με ποιους τρόπους μπορούμε να σχεδιάσουμε ολοκληρωμένα κυκλώματα που να είναι ανθεκτικά σε αυτά τα φαινόμενα.

Στο δεύτερο κεφάλαιο θα δούμε τι είναι ο χαρακτηρισμός και τι πληροφορίες παίρνουμε για το κάθε κελί, όπου κελί είναι κάποια πύλη, ένα flip-flop, ένας αθροιστής κλπ., θα δούμε τι είναι τα ακρότατα σημεία και τα μοντέλα χρονισμού και ισχύος που χρησιμοποιούνται στον χαρακτηρισμό και πως παρουσιάζονται στο πρότυπο Liberty που είναι το αρχείο (.lib) που περιλαμβάνει όλες τις πληροφορίες του χαρακτηρισμού. Επίσης θα γίνει αναφορά στα κελιά που θα χαρακτηριστούν και στη διαδικασία χαρακτηρισμού και τέλος θα γίνει σύγκριση των αποτελεσμάτων με τα πρωτότυπα κελιά.

Στο τρίτο κεφάλαιο αρχικά θα δούμε τι είναι το πρωτόκολλο επικοινωνίας SPI και η μέθοδος σχεδίασης με χρήση τριπλού πλεονασμού (TMR). Στη συνέχεια θα δούμε λεπτομέρειες για τη σύνθεση και τη φυσική σχεδίαση δύο υλοποιήσεων του slave του SPI. Μία υλοποίηση με χρήση απλού TMR, δηλαδή τριπλασιασμό μόνο των καταχωρητών και μία υλοποίηση με χρήση πλήρους TMR, δηλαδή τριπλασιασμό καταχωρητών, συνδυαστικής λογικής, εισόδων και εξόδων. Τέλος θα δούμε τις προσομοιώσεις και τα αποτελέσματα της κάθε υλοποίησης και θα γίνει σύγκριση των αποτελεσμάτων των δύο υλοποιήσεων, καθώς και σύγκριση με χρήση δακτυλίων προστασίας και χρήση κανονικών κελιών.

Στο τέταρτο κεφάλαιο θα δούμε συνοπτικά τα αποτελέσματα και θα αναφέρουμε τα συμπεράσματα της παρούσας εργασίας.

Τα εργαλεία που χρησιμοποιήθηκαν είναι:

- Για τον χαρακτηρισμό: Liberate v15.10 της Cadence.
- Για την σύνθεση: Genus v15.20 της Cadence.
- Για την φυσική σχεδίαση (layout): Innovus v15.20 της Cadence.
- Για την προσομοίωση: Simvision v15.10 της Cadence.

1. ΕΙΣΑΓΩΓΗ

1.1 Περιβάλλον διαστημικής ακτινοβολίας

Η Γη προστατεύεται από την ιονίζουσα ακτινοβολία με προέλευση το διάστημα από την ατμόσφαιρα, η οποία ενεργεί ως ημιδιαπερατό "κόσκινο". Αφήνει όλο το φως και τη θερμότητα να περάσουν, ενώ σταματά σε σημαντικό βαθμό την ακτινοβολία αυτή. Επομένως σε ηλεκτρονικά κυκλώματα που λειτουργούν σε συνθήκες μειωμένης προστασίας από την ιονίζουσα ακτινοβολία, πρέπει να γίνει ειδική σχεδίαση με χρήση ενός συνόλου από περιορισμούς, ώστε να είναι σε θέση να λειτουργήσουν αξιόπιστα.

Οι κύριες πηγές της ακτινοβολίας που εντοπίζονται στο διάστημα, ταξινομούνται σε τέσσερις κατηγορίες ως προς την προέλευση τους: Ζώνες ακτινοβολίας Βαν Άλεν, ηλιακές εκρήξεις, ηλιακός άνεμος και κοσμικές ακτίνες [1]. Τα σωματίδια που μας ενδιαφέρουν είναι ηλεκτρόνια, πρωτόνια και ιόντα διαφορετικής προέλευσης, και με διάφορες ενέργειες. Όλα τα σωματίδια συμβάλλουν στην υποβάθμιση της απόδοσης των ηλεκτρονικών κυκλωμάτων λόγω των φαινομένων που οφείλονται στην συνολική δόση (συνολική ενέργεια που απορροφάται ανά μονάδα μάζας), όμως τα βαρέα ιόντα και πρωτόνια προκαλούν έναν αριθμό από ειδικά φαινόμενα που ομαδοποιούνται και ονομάζονται επιδράσεις απλού συμβάντος (Single Event Effects – SEE).

1.1.1 Ζώνες ακτινοβολίας Βαν Άλεν

Οι ζώνες Βαν Άλεν περιέχουν παγιδευμένα ηλεκτρόνια και πρωτόνια. Υπάρχουν δύο ζώνες ηλεκτρονίων. Η εσωτερική ζώνη περιέχει ηλεκτρόνια των οποίων η ενέργεια είναι μικρότερη από 5 MeV. Η εξωτερική ζώνη περιέχει ηλεκτρόνια των οποίων η ενέργεια μπορεί να φθάσει τα 7 MeV, επιπλέον στην περίπτωση της εξωτερικής ζώνης, η ροή ηλεκτρονίων είναι και πιο μεταβλητή και περισσότερο έντονη από αυτή της εσωτερικής ζώνης. Μια τρίτη ζώνη ηλεκτρονίων παρατηρήθηκε μετά από μαγνητική καταιγίδα στις 24 Μαρτίου 1991. Αυτή η ζώνη βρίσκεται ανάμεσα στην πρώτη δεύτερη ζώνη και η ενέργεια των παγιδευμένων ηλεκτρονίων της, φθάνει μέχρι και 30 MeV.

Η παγιδευμένη ακτινοβολία περιλαμβάνει επίσης μια ζώνη πρωτονίων των οποίων η ενέργεια είναι μεγαλύτερη από 50 MeV [2]. Μια δεύτερη τέτοια ζώνη, που περιέχει πρωτόνια υψηλής ενέργειας (>100 MeV), εμφανίστηκε μετά από την ίδια μαγνητική καταιγίδα που αναφέρθηκε παραπάνω. Συνεπώς, μια βίαιη μαγνητική καταιγίδα μπορεί να δημιουργήσει νέες ζώνες ακτινοβολίας των οποίων το προσδόκιμο ζωής υπολογίζεται σε περισσότερα από δύο χρόνια, αλλά δεν είναι γνωστό με ακρίβεια.

1.1.2 Ηλιακές εκλάμψεις

Ο 11χρονος κύκλος ηλιακών κηλίδων μπορεί να υποδιαιρεθεί σε τέσσερα έτη χαμηλής δραστηριότητας και επτά υψηλής δραστηριότητας. Υπάρχουν δύο τύποι γεγονότων που πρέπει να ληφθούν υπόψη σε σχέση με το περιβάλλον ακτινοβολίας που εξετάζεται εδώ. Ο πρώτος είναι η στεμματική εκπομπή μάζας, η οποία διαρκεί αρκετές ημέρες και εκπέμπει κυρίως πρωτόνια υψηλής ενέργειας (έως λίγες εκατοντάδες MeV). Το σημείο αναφοράς εδώ είναι η έκλαμψη πρωτονίων που έλαβε χώρα τον Αύγουστο του 1972. Αυτό το συμβάν καταγράφηκε την περίοδο του 20ου ηλιακού κύκλου και το 84% των ηλιακών πρωτονίων που παρείχε, είχαν ενέργεια πάνω από 30 MeV. Ο δεύτερος τύπος γεγονότων εμπίπτει στη κατηγορία των "παρορμητικών συμβάντων" που συνεπάγονται μεγάλες εκπομπές βαριών ιόντων. Αυτές περιλαμβάνουν ηλιακές εκλάμψεις με υψηλή ενέργεια (από μερικές δεκάδες MeV έως αρκετές εκατοντάδες GeV ανά νουκλεόνιο). Η

αναφορά εδώ είναι οι εκλάμψεις βαριών ιόντων του Σεπτεμβρίου του 1977 και της 24^{ης} Οκτωβρίου του 1989.

1.1.3 Ηλιακός άνεμος

Η υψηλή θερμοκρασία του ηλιακού στέμματος συνεισφέρει αρκετή ενέργεια για να επιτρέψει στα ηλεκτρόνια να ξεφύγουν από τη βαρυτική έλξη του ήλιου. Η επίδραση της εκτόξευσης ηλεκτρονίων είναι μια ανισορροπία φορτίου που έχει ως αποτέλεσμα την εκτόξευση πρωτονίων και βαρύτερων ιόντων από το στέμμα. Το εκτοξευόμενο αέριο είναι τόσο καυτό που τα σωματίδια είναι ομογενοποιημένα σε αραιό πλάσμα. Η ενεργειακή πυκνότητα του πλάσματος υπερβαίνει την πυκνότητα του μαγνητικού του πεδίου, και έτσι το ηλιακό μαγνητικό πεδίο είναι «παγωμένο» μέσα στο πλάσμα. Αυτό το ηλεκτρικά ουδέτερο πλάσμα ρέει από τον ήλιο προς τα έξω με ταχύτητες περίπου 300 έως 900 χιλιόμετρα ανά δευτερόλεπτο με θερμοκρασίες από 10⁴ έως 10⁶ C°.

Σημαντικές διαταραχές στο γεωμαγνητικό πεδίο μπορεί να συμβούν με αλλαγές στην πυκνότητα του ηλιακού ανέμου (π.χ. ηλιακή έκλαμψη), στη ταχύτητα του ηλιακού ανέμου (π.χ. στεμματική εκπομπή μάζας), και στον προσανατολισμό του ενσωματωμένου ηλιακού μαγνητικού πεδίου. Η στεμματική εκπομπή μάζας και οι ηλιακές εκλάμψεις προκαλούν διαταραχές του ηλιακού ανέμου. Η αλληλεπίδραση μεταξύ αυτών των διαταραχών και της μαγνητόσφαιρας της γης, προκαλεί τις μαγνητικές καταιγίδες. Ο συσχετισμός του αριθμού των καταιγίδων με το επίπεδο της ηλιακής δραστηριότητας είναι πολύ ισχυρός και οι κύριες μαγνητικές καταιγίδες συνδέονται στενά με την στεμματική εκπομπή μάζας. Κατά τη διάρκεια της περιόδου υψηλής ηλιακής δραστηριότητας, παρατηρούνται διακυμάνσεις στο μαγνητικό πεδίο του ήλιου, και αυτές οι διακυμάνσεις έχουν σαν αποτέλεσμα τη συμπίεση των γραμμών του μαγνητικού πεδίου της γης (γεωμαγνητικές καταιγίδες). Όταν η συμπίεση των γραμμών του μαγνητικού πεδίου της γης εμφανίζεται, το πλάσμα στη νυχτερινή πλευρά της γης ωθείται προς την επιφάνεια της γης. Καθώς αυτό το πλάσμα ωθείται πιο κοντά στη γη, τα ηλεκτρόνια και τα ιόντα εκτρέπονται από το γήινο μαγνητικό πεδίο.

1.1.4 Κοσμικές ακτίνες

Οι κοσμικές ακτίνες είναι πυρήνες βαρέων ιόντων (χωρίς το περιβάλλον σύννεφο ηλεκτρονίων). Στην πραγματικότητα, αυτά τα βαρέα ιόντα αντιπροσωπεύουν μόνο το 1% της κοσμικής ακτινοβολίας, το οποίο κατά τα άλλα περιέχει 83% πρωτόνια, 13% πυρήνες ηλίου και 3% ηλεκτρόνια. Η προέλευση αυτής της ακτινοβολίας δεν έχει πραγματικά διευκρινιστεί. Ωστόσο, γνωρίζουμε ότι ένα μέρος της (δηλαδή τα πιο ενεργητικά ιόντα) προέρχονται από το εξωτερικό του Γαλαξία μας και τα υπόλοιπα μέσα από αυτόν. Οι ενέργειες των ιόντων είναι πολύ υψηλές (το πιο ενεργητικό ιόν που ανιχνεύθηκε ποτέ είχε ενέργεια 3·10²⁰ eV, δηλαδή σχεδόν 50 J!) και οι μηχανισμοί επιτάχυνσης δεν είναι πάντα καλά κατανοητοί. Η κοσμική ακτινοβολία είναι σχεδόν ισοτροπική όταν φτάνει κοντά στην μαγνητόσφαιρα. Ωστόσο, επειδή η ακτινοβολία συμπλέκεται με το μαγνητικό πεδίο της Γης, η ισοτροπία της δεν διατηρείται. Ενώ η σύνθεσή της είναι σχεδόν ίδια με εκείνη της ύλης που βρίσκεται στον τοπικό γαλαξία, φαίνεται να επηρεάζεται από την αλληλεπί-δραση με τη διαστρική ύλη. Σε ενέργειες μικρότερες από 1 GeV ανά νουκλεόνιο, η ροή σωματιδίων εξαρτάται από την ηλιακή δραστηριότητα.

1.2 Επιδράσεις απλού συμβάντος στις ηλεκτρονικές συσκευές και συστήματα

Οι επιδράσεις απλού συμβάντος (SEEs) προκαλούνται από την αλληλεπίδραση ενός ιονίζοντος σωματιδίου με ηλεκτρονικά εξαρτήματα [3]. Τα ιονισμένα σωματίδια μπορούν να είναι πρωτογενή (όπως βαρέα ιόντα στο περιβάλλον του διαστήματος ή σωματίδια

άλφα που παράγονται από ραδιενεργά ισότοπα που περιέχονται στη ψηφίδα του ολοκληρωμένου κυκλώματος) ή δευτερογενή που δημιουργούνται από την πυρηνική αλληλεπίδραση ενός σωματιδίου, όπως ένα νετρόνιο ή ένα πρωτόνιο με πυρίτιο, οξυγόνο ή οποιοδήποτε άλλο άτομο της ψηφίδας.

Σήμερα, οι επιπτώσεις αυτές θεωρούνται σημαντική πρόκληση, οι οποίες περιορίζουν έντονα την αξιοπιστία και τη διαθεσιμότητα των ηλεκτρονικών συστημάτων και γίνονται πολλές προσπάθειες έρευνας και ανάπτυξης τόσο στη βιομηχανία, όσο και στον ακαδημαϊκό κόσμο.

1.2.1 Προσδιορισμός επιδράσεων απλού συμβάντος

Οι επιδράσεις απλού συμβάντος στις ηλεκτρονικές συσκευές ομαδοποιούν όλες τις πιθανές επιδράσεις που προκαλεί η αλληλεπίδραση ενός πυρηνικού σωματιδίου με ηλεκτρονικά εξαρτήματα. Αυτές οι επιδράσεις ταξινομούνται σε δύο κατηγορίες. Τις κατάστροφικές επιδράσεις, που είναι τα σφάλματα υλικού (hard errors) και τις μη καταστροφικές επιδράσεις, που είναι τα λογικά σφάλματα (soft errors). Τα λογικά σφάλματα μπορούν να αποκατασταθούν με επαναφορά (reset), ή απλά με επανεγγραφή των πληροφοριών.

Στις καταστροφικές επιδράσεις ανήκουν τα παρακάτω:

Το κάψιμο εξαιτίας απλού συμβάντος (Burnout Single Event - SEB) και η ρήξη πύλης απλού συμβάντος (Single Event Gate Rupture - SEGR) είναι δύο κύριοι τύποι συμβάντων με καταστροφικές συνέπειες και αφορούν τα τρανζίστορ επίδρασης πεδίου από ημιαγωγό μεταλλικού οξειδίου (Metal Oxide Semiconductor Field Effect Transistor - MOSFET) ισχύος (μεγάλος λόγος πλάτους/μήκους της πύλης).

SEGR ορίζεται ως ένα συμβάν, όπου το ρεύμα πύλης και το ρεύμα απαγωγού αυξάνονται απότομα κατά τη διάρκεια της ακτινοβολίας ή μετά, ενώ το τρανζίστορ βρίσκεται σε κατάσταση OFF. Η SEGR δεν επηρεάζεται από τη θερμοκρασία και είναι πιο πιθανό να συμβεί όταν το ιόν πέσει σχεδόν κάθετα με τον κατακόρυφο άξονα της διάταξης.

SEB ορίζεται ως ένα συμβάν στο οποίο το φορτίο που παράγεται από την ενέργεια που εναποθέτει το ιόν στο πυρίτιο σε συνδυασμό με το υπάρχον ηλεκτρικό πεδίο στο εσωτερικό της ηλεκτρονικής διάταξης, δημιουργεί ικανό ρεύμα για ικανό χρόνο ώστε να προκαλέσει καταστροφική υπερθέρμανση ή διάρρηξη της διάταξης.

Τα παραπάνω συμβάντα δεν θα μας απασχολήσουν στην παρούσα εργασία και δεν θα αναφερθούμε περισσότερο. [4] [5]

Το κλείδωμα απλού συμβάντος (Single Event Latch-up - SEL). Στη τεχνολογία συμπληρωματικού ημιαγωγού μεταλλικού οξειδίου (Complementary Metal-Oxide Semiconductor - CMOS), οι παρασιτικές δομές PNPN μπορεί να ενεργοποιηθούν δίνοντας το SEL, το οποίο συνδέεται με μια ισχυρή αύξηση του ρεύματος τροφοδοσίας και ίσως έχει καταστροφικές συνέπειες λόγω υπερβολικής θέρμανσης στη δομή και στα μέταλλα του ημιαγωγού. Το SEL για να απενεργοποιηθεί χρειάζεται διακοπή της τροφοδοσίας ρεύματος.

Στις μη καταστροφικές επιδράσεις ανήκουν τα παρακάτω:

Η μετάβαση απλού συμβάντος (Single Event Transient - SET) στα αναλογικά κυκλώματα είναι κυρίως μεταβατικοί παλμοί σε τελεστικούς ενισχυτές, συγκριτές ή κυκλώματα τάσης αναφοράς. Στην συνδυαστική λογική, SET είναι παλμοί που παράγονται σε μια πύλη και μπορεί να διαδοθούν σε ένα μονοπάτι συνδυαστικού κυκλώματος και τελικά να μανταλωθούν σε ένα στοιχείο μνήμης (π.χ. flip-flop).

Στα στοιχεία μνήμης (μανταλωτές, καταχωρητές) το απλό συμβάν ονομάζεται **Διατάραξη** απλού συμβάντος (Single Event Upset - SEU). Αυτό αντιστοιχεί σε μια αναστροφή της κατάστασης του στοιχείου μνήμης. Όταν η διέλευση ενός σωματιδίου επηρεάζει πολλά flip-flops, δηλαδή αν περισσότερα από ένα bit σε μια λέξη αλλάξουν, τότε έχουμε πολυκύτταρη διαταραχή (Multi-Cell Upset - MCU).

1.2.2 Κλείδωμα απλού συμβάντος (SEL)

Το κλείδωμά απλού συμβάντος (SEL) [3],[6] μπορεί να είναι ένα μόνιμο σφάλμα και κατάστροφικό για τη διάταξη. Εξαιτίας της χρήσης του πηγαδιού τύπου Ν για την δημιουργία του τρανζίστορ τύπου Ρ δημιουργούνται δύο παρασιτικά διπολικά τρανζίστορ, ένα PNP και ένα NPN τα οποία σε συνδυασμό σχηματίζουν μια διάταξη PNPN. Στο σχήμα 1.1 φαίνεται η διατομή ενός CMOS αντιστροφέα και στο σχήμα 1.2 το ισοδύναμο του.

Η επαφή Ν-πηγαδιού / Ρ-υποστρώματος είναι η επαφή βάσης-συλλέκτη και για τα δύο τρανζίστορ. Μια ιδανική δομή PNPN είναι ενεργή, δηλαδή έχουμε SEL, αν το γινόμενο του κέρδους των NPN και PNP τρανζίστορ είναι β₁ x β₂ > 1, όπου β είναι το κέρδος ρεύματος (ρεύμα συλλέκτη προς ρεύμα βάσης) του διπολικού τρανζίστορ.



Σχήμα 1.1: Βασικός μηχανισμός SEL σε έναν αντιστροφέα CMOS. [3]

Εάν περάσει αρκετό ρεύμα μέσω της R_{sub} και ενεργοποιηθεί το Q2, (I * R_{sub} > 0.7V), αυτό με τη σειρά του θα αντλήσει ρεύμα μέσω της R_{well}. Εάν η πτώση τάσης στην R_{well} είναι αρκετά υψηλή, το Q1 θα ενεργοποιηθεί και έτσι θα δημιουργηθεί ένα μονοπάτι χαμηλής αντίστασης μεταξύ τροφοδοσίας και γείωσης. Αν το β_{oλ} > 1, όπως αναφέρθηκε προηγουμένως, τότε λόγω του υψηλού ρεύματος που θα υπάρξει, αναπτύσσεται μεγάλη θερμοκρασία και τα μέταλλα μέσα στο ολοκληρωμένο λιώνουν. Από τη στιγμή που συμβεί κλείδωμα, παραμένει έως ότου μειωθεί το ρεύμα και αυτό γίνεται με την διακοπή της παροχής τροφοδοσίας στο ολοκληρωμένο, διαφορετικά το ολοκληρωμένο θα καταστραφεί.

Ο μηχανισμός ενεργοποίησης μπορεί να γίνει κατανοητός λαμβάνοντας υπόψη το ισοδύναμο σχήμα με τα διασυνδεδεμένα διπολικά τρανζίστορ NPN και PNP (σχήμα 1.2). Η δομή PNPN έχει δύο σταθερές καταστάσεις: Ανενεργή κατάσταση, όπου και τα δύο διπολικά τρανζίστορ είναι απενεργοποιημένα και ενεργή κατάσταση, όπου και τα δύο διπολικά τρανζίστορ είναι ενεργοποιημένα. Η ελάχιστη τάση και ρεύμα που απαιτείται για

τη διατήρηση της ενεργής κατάστασης, καλούνται αντίστοιχα διατηρούσα τάση και διατηρών ρεύμα.



Σχήμα 1.2: Ισοδύναμη διάταξη της PNPN παρασιτικής δομής. [3]

Η διατηρούσα τάση πρέπει να συγκρίνεται με την τάση τροφοδοσίας. Όταν η δομή PNPN είναι ενεργοποιημένη, και οι δύο επαφές βάσης-εκπομπού και συλλέκτη-βάσης είναι ενεργοποιημένες. Έτσι, η ελάχιστη διατηρούσα τάση μεταξύ Vdd και GND είναι περίπου 0,6V. Στους περισσότερους προηγμένους τεχνολογικούς κόμβους, η διατηρούσα τάση είναι σχεδόν ίση ή μεγαλύτερη από την τάση τροφοδοσίας. Έτσι, η μείωση της τάσης τροφοδοσίας περιορίζει την εμφάνιση του SEL (Σχήμα 1.3).



Σχήμα 1.3: Ενεργοποίηση PNPN δομής. Το ιόν διασχίζει την επαφή Ν-Πηγαδιού / Ρυποστρώματος. Το ρεύμα ενισχύθηκε στη δομή θετικής ανάδρασης. [3]

Η χαρακτηριστική ρεύματος τάσης μια τέτοιας διάταξης φαίνεται στο σχήμα 1.4 [7]. Όπως φαίνεται για μία τιμή στην τάση αντιστοιχούν παραπάνω από μία τιμές ρεύματος. Η καμπύλη μπορεί να χωριστεί σε δύο περιοχές. Στην πρώτη περιοχή (περιοχή Ι), που αντιστοιχεί και στην κανονική λειτουργία, το ρεύμα είναι χαμηλό για όλες τις τιμές της τάσης. Στην περίπτωση που συμβεί SEL η διάταξη περνάει στην περιοχή ΙΙ όπου για χαμηλή τάση έχουμε υψηλό ρεύμα. Εάν το κέρδος της συνολικής διάταξης είναι μεγαλύτερο από ένα, τότε η κλειδωμένη κατάσταση μπορεί να διατηρηθεί από μόνη της, έως ότου διακοπεί η τάση τροφοδοσίας ή καταστραφεί η συσκευή.



Σχήμα 1.4: Καμπύλη ρεύματος - τάσης για την διάταξη pnpn. [6]

Εάν μία διάταξη πρόκειται να χρησιμοποιηθεί σε διαστημική εφαρμογή, τότε πρέπει να ληφθούν ειδικά μέτρα, όπως η ύπαρξη αντιστάσεων περιορισμού του ρεύματος ή η δυνατότητα να διακόπτεται η τάση τροφοδοσίας της διάταξης στην περίπτωση που ανιχνευθεί SEL [8]. Το τελευταίο αν και πολύ αποτελεσματικό, σε πολλές περιπτώσεις δεν είναι επιθυμητό. Η διάταξη θα πρέπει να είναι ανθεκτική σε SEL προκειμένου να είναι εφικτή η χρησιμοποίησή τη στο διαστημικό περιβάλλον.

1.2.3 Μη καταστροφικές επιδράσεις

Όπως απεικονίζεται στο σχήμα 1.5, όταν ένα σωματίδιο προσκρούει πάνω σε μία ημιαγωγική διάταξη, απελευθερώνει ζεύγη ηλεκτρονίων-οπών. Τα ζεύγη αυτά απουσία κάποιου ηλεκτρικού πεδίου επανασυνδέονται μετά από κάποιο χρονικό διάστημα. Στην περίπτωση παρουσίας ηλεκτρικού πεδίου (όπως στην περίπτωση μίας ανάστροφα πολωμένης διόδου) πολλά από τα φορτία θα συλλεχθούν στους δύο ακροδέκτες με συνέπεια την δημιουργία ενός παλμού ρεύματος για ένα μικρό χρονικό διάστημα [6] [9].



Σχήμα 1.5: Σχηματισμός ζευγών ηλεκτρονίων οπών κατά την πρόσκρουση του σωματιδίου πάνω στο υλικό. [6]

Στην περίπτωση που ο κόμβος στον οποίο προσπίπτει το σωματίδιο είναι ο απαγωγός ενός τρανζίστορ (σχήμα 1.6) το οποίο είναι σε κατάσταση αποκοπής, το επαγόμενο ρεύμα είναι ικανό να φορτίσει / αποφορτίσει τη χωρητικότητα του κόμβου δημιουργώντας μια αιχμή στην τάση. [10]



Σχήμα 1.6: Σωματίδιο προσκρούει πάνω στον απαγωγό ενός τρανζίστορ και παράγει μια αιχμή στο ρεύμα εξόδου. [6]

1.2.3.1 Διατάραξη απλού συμβάντος (SEU)

Όταν ένα χτύπημα ιονισμένου σωματιδίου τροποποιεί την ηλεκτρική κατάσταση ενός στοιχείου μνήμης (flip-flop, μανταλωτής, κλπ), έτσι ώστε να προκύπτει σφάλμα κατά την ανάγνωση του στοιχείου αυτού, τότε έχουμε SEU. Στην περίπτωση που έχουμε συνδεσμολογία μνήμης, στο κύκλωμά της υπάρχει ανάδραση η οποία κρατάει την πληροφορία. Η βασική δομή ενός κυκλώματος μνήμης φαίνεται στο σχήμα 1.7. Αποτελείται από δύο αντιστροφείς με την έξοδο του ενός να είναι συνδεδεμένη στην είσοδο του άλλου. Υποθέτουμε ότι ο αντιστροφέας U1 βρίσκεται σε λογικό '1' και ο αντιστροφέας U2 βρίσκεται σε λογικό '0'. Υποθέτουμε επίσης ότι ένα φορτισμένο σωματίδιο προσκρούει στον απαγωγό του τρανζίστορ Mn1. Όπως είπαμε και παραπάνω η πρόσκρουση θα απελευθερώσει ένα φορτίο στον κόμβο το οποίο όταν θα συλλεχθεί θα προκαλέσει ένα βύθισμα στην τάση εξόδου του U1, το μέγεθος του οποίου θα εξαρτάται από την ενέργεια, τη μάζα και τη σημείο πρόσκρουσης του σωματιδίου. Στην περίπτωση που η ποσότητα του φορτίου το οποίο συλλέχθηκε είναι μεγαλύτερη από μία κρίσιμη τιμή, το βύθισμα θα είναι ικανό να αλλάξει κατάσταση στον αντιστροφέα U2 και έτσι η πληροφορία θα χαθεί [11] [12]. Το φαινόμενο αυτό αποτελεί μία από τις κυριότερες αιτίες που οι εμπορικοί μικροεπεξεργαστές δεν μπορούν να χρησιμοποιηθούν για διαστημικές εφαρμογές [13].



Σχήμα 1.7: Απλουστευμένο διάγραμμα μιας μονάδας μνήμης. [6]

Σε μία στατική μνήμη τυχαίας προσπέλασης (Static Random Access Memory - SRAM) ή ένα flip-flop, η κατάσταση της μνήμης αντιστρέφεται. Σε μια δυναμική μνήμη τυχαίας προσπέλασης (Dynamic Random Access Memory - DRAM), η αποθηκευμένη φόρτιση μπορεί να είναι ελαφρώς τροποποιημένη και ερμηνευμένη ως λανθασμένη τιμή από το κύκλωμα ανάγνωσης. Στα λογικά κυκλώματα, οι SEUs μπορούν επίσης να προκληθούν από ένα SET όταν διαδίδεται μέσω της συνδυαστικής λογικής και αποθηκεύεται από ένα στοιχείο μνήμης.

Αυτού του είδους το σφάλμα είναι προσωρινό και φεύγει την επόμενη φορά που ξαναφορτώνουμε την μνήμη. Είναι όμως προφανές ότι εάν η συχνότητα των SEU είναι πολύ μεγάλη τότε δεν είναι δυνατή η πρακτική χρήση του ολοκληρωμένου. Είναι προφανές ότι όσο μικρότερη είναι η χωρητικότητα στον κόμβο στον οποίο προσκρούει το σωματίδιο, τόσο μικρότερο θα είναι το φορτίο το οποίο θα είναι ικανό να προκαλέσει αντιστροφή. Με τη σμίκρυνση που υπάρχει σήμερα στα ολοκληρωμένα κυκλώματα, η κρίσιμη ποσότητα φορτίου η οποία να είναι ικανή να προκαλέσει SEU έχει μειωθεί πολύ. Η υψηλή ταχύτητα και η πολυπλοκότητα των σύγχρονων ολοκληρωμένων κυκλωμάτων επιτεύχθηκε μειώνοντας το μέγεθος των τρανζίστορ και συνεπώς την ενέργεια αποθήκευσης του κάθε bit. Η μειωμένη τάση τροφοδοσίας στην οποία είναι επιθυμητό να λειτουργούν σήμερα πολλές διατάξεις, τις κάνουν περισσότερο ευαίσθητες σε SEU.

1.2.3.2 Μετάβαση απλού συμβάντος (SET)

Οι μεταβάσεις απλού συμβάντος στα ψηφιακά κυκλώματα είναι προσωρινές διαταραχές τάσης ή ρεύματος που επηρεάζουν τις μεταβάσεις στις εξόδους των πυλών. Παρόλο που ένα SET προκαλεί μια μετάβαση στην έξοδο της πύλης που προσκρούει το ιόν, αυτή η μετάβαση μπορεί να μεταδοθεί μέσω των επόμενων πυλών και τελικά να προκαλέσει μια SEU όταν φτάσει σε ένα στοιχείο μνήμης. Για να οδηγηθεί σε σφάλμα κυκλώματος μία SET πρέπει να πληρούνται τέσσερα κριτήρια. [14]

- Η πρόσκρουση του σωματιδίου πρέπει να δημιουργήσει μια μετάβαση ικανή να μεταδοθεί μέσω του κυκλώματος.

- Πρέπει να υπάρχει ένα ανοικτό λογικό μονοπάτι μέσω του οποίου μπορεί να διαδοθεί η SET και να φθάσει σε ένα flip flop ή άλλο στοιχείο μνήμης.

- Η SET πρέπει να έχει επαρκές εύρος και διάρκεια για να αλλάξει την κατάσταση του ένα στοιχείο μνήμης.

 Σε κυκλώματα σύγχρονη λογικής, η SET πρέπει να φτάσει στο στοιχείο μνήμης, στην αιχμή του ρολογιού κατά την οποία αλλάζει κατάσταση το στοιχείο μνήμης και εντός του χρονικού παραθύρου που ορίζουν οι χρόνοι αποκατάστασης και συγκράτησης.

Η πιθανότητα ότι τα σφάλματα στις μεταβάσεις θα ληφθούν ως έγκυρα δεδομένα σε ένα κύκλωμα συνδυαστικής λογικής αυξάνεται γραμμικά με τη συχνότητα επειδή αυξάνεται η συχνότητα των αιχμών του ρολογιού. Καθώς οι ταχύτητες του κυκλώματος αυξάνονται, αυξάνεται και η δυνατότητα διάδοσης των μεταβάσεων. Υψηλή ταχύτητα σημαίνει ταχύτερες πύλες και / ή λιγότερα λογικά επίπεδα σε κάθε στάδιο σωλήνωσης (pipeline). Ωστόσο, θα μπορούσαμε επίσης να υποθέσουμε ότι η διάρκεια της κάθε μετάβασης μειώνεται. Λόγω της ταχύτερης διάδοσης τους σε υψηλής ταχύτητας κυκλώματα και την μεγαλύτερη πιθανότητα αποθήκευσής τους από τα στοιχεία μνήμης, όπως τα flip-flops, έχει προβλεφθεί ότι η SET θα αποτελέσει ένα πολύ σημαντικό θέμα στα ψηφιακά ολοκληρωμένα κυκλώματα (ICs) [15] [16].

1.3 Επιδράσεις ολικής δόσης ιονισμού (Total lonizing dose - TID)

Στην ενότητα αυτή θα δούμε πως το φορτίο το οποίο συσσωρεύεται από τον ιονισμό επιδράει και αλλάζει τα χαρακτηριστικά του MOSFET. Η πιο σημαντική επίδραση είναι η αλλαγή της τάσης κατωφλίου του τρανζίστορ και κατά συνέπεια της δυνατότητας οδήγησης αυτού.

Όταν από την ακτινοβολία σχηματίζεται ένα ζεύγος ηλεκτρονίου-οπής, το ηλεκτρόνιο αλλάζει ενεργειακή ζώνη και πηγαίνει από τη ζώνη σθένους στη ζώνη αγωγιμότητας, αφήνοντας πίσω του μια οπή. Λόγω ύπαρξης ηλεκτρικού πεδίου, τα ηλεκτρόνια παρασύρονται πολύ εύκολα αφού η κινητικότητα τους είναι μεγαλύτερη από αυτή των οπών. Αν

εξαιρέσουμε ένα μικρό αριθμό ζευγών που επανασυνδέονται πολύ γρήγορα, τα υπόλοιπα ηλεκτρόνια και οπές είναι ελεύθερα να κινηθούν στο υλικό μέχρι να επανασυνδεθούν, η να παγιδευτούν. Τα παγιδευμένα φορτία δημιουργούν δικό τους ηλεκτρικό πεδίο, το οποίο με την σειρά του αλλοιώνει τα χαρακτηριστικά του ημιαγωγού [6].

Το πρόβλημα της συσσώρευσης παγιδευμένων φορτίων έχει γίνει αντικείμενο εντατικής μελέτης για τις διατάξεις MOS (Metal Oxide Semiconductor), αφού πάνω σε αυτή την τεχνολογία είναι βασισμένο το μεγαλύτερο τμήμα των μικροηλεκτρονικών κυκλωμάτων. Το επαγόμενο πεδίο από την αποθήκευση σωματιδίων στο οξείδιο, έχει ως αποτέλεσμα την αύξηση των ρευμάτων διαρροής, καθώς επίσης και τη μεταβολή στην τάση κατωφλίου του τρανζίστορ [17], [18].

Στο σχήμα 1.8 απεικονίζεται μία αναπαράσταση ενός MOSFET τρανζίστορ τύπου n, το οποίο χρησιμοποιεί υπόστρωμα τύπου p. Όταν στην πύλη του τρανζίστορ ασκηθεί μία τάση, θα παραχθεί ένα ηλεκτρικό πεδίο κάθετο προς το οξείδιο το οποίο βρίσκεται κάτω από την πύλη. Εάν αυτή η τάση είναι θετική και αρκετά μεγάλη, τότε οι φορείς πλειονότητας (οπές) θα απομακρυνθούν από την περιοχή που βρίσκεται κάτω από την πύλη και σε αυτή θα προσελκυσθούν οι φορείς μειονότητας, φτιάχνοντας το κανάλι αντιστροφής. Εάν εφαρμοστεί μία τάση ανάμεσα στον απαγωγό και στην πηγή, τότε το κανάλι αντιστροφής. Εάν εφαρμοστεί μία σύνδεση χαμηλής αντίστασης προκειμένου τα ηλεκτρόνια να κινηθούν από την πύλη που χρειάζεται ώστε να μπορεί να περάσει σημαντική ποσότητα ρεύματος απ' το κανάλι, λέγεται τάση κατωφλίου (threshold voltage).



Σχήμα 1.8: Σχηματικό διάγραμμα ενός MOSFET τύπου-η σε κανονική λειτουργία. [6]

Κατά την ακτινοβολία παγιδεύονται θετικά φορτία στο διηλεκτρικό (διοξείδιο του πυριτίου) το οποίο βρίσκεται κάτω από την πύλη. Τα παγιδευμένα αυτά φορτία προκαλούν παραμένον ηλεκτρικό πεδίο στην επιφάνεια του πυριτίου κάτω από την πύλη, το οποίο με την σειρά του προκαλεί ελάττωση της τάσης κατωφλίου της διάταξης. Για μεγάλες συγκεντρώσεις φορτίου, η διάταξη μπορεί να είναι σε κατάσταση λειτουργίας, ακόμη και για μηδενική τάση στην πύλη. Η παραπάνω κατάσταση απεικονίζεται στο σχήμα 1.9.



Σχήμα 1.9: Σχηματικό διάγραμμα ενός MOSFET τύπου-η μετά την ακτινοβολία. [6]

1.3.1 Αύξηση των παρασιτικών ρευμάτων

Στις σύγχρονες τεχνολογίες όπου το οξείδιο του πυριτίου γίνεται ολοένα και πιο λεπτό, το πρόβλημα της μετατόπισης της τάσης κατωφλίου γίνεται πιο μικρό.

Όπως φαίνεται στο σχήμα 1.11 υπάρχουν και δύο παρασιτικά τρανζίστορ στις άκρες του κανονικού τα οποία δημιουργούνται από την πηγή, τον απαγωγό και από οξείδιο του πεδίου. Το οξείδιο του πεδίου είναι συνήθως πιο παχύ από το οξείδιο της πύλης και γι αυτό το λόγο με την ακτινοβολία παρατηρείται μεγάλη συσσώρευση παγιδευμένου θετικού φορτίου σε αυτή την περιοχή. Έτσι, είναι δυνατόν μετά την ακτινοβολία τα παρασιτικά αυτά τρανζίστορ να συνεισφέρουν στο ρεύμα διαρροής σε τέτοιο βαθμό, που γίνεται η αιτία της καταστροφής του τρανζίστορ [19] [20] [21]. Στο σχήμα 1.10 απεικονί-ζεται η κάτοψη του τρανζίστορ προκειμένου να φανεί καλύτερα το ρεύμα διαρροής.



Σχήμα 1.10: Κάτοψη ενός τρανζίστορ όπου φαίνονται και τα παρασιτικά ρεύματα. [6]



Σχήμα 1.11: Δημιουργία των παρασιτικών ρευμάτων από την πηγή στον απαγωγό. [6]

1.4 Ανθεκτικότητα σε επιδράσεις ολικής δόσης

Στην ενότητα αυτή θα δούμε κάποιες τεχνικές με τις οποίες μπορούμε να πετύχουμε ανθεκτικότητα στις συσσωρευτικές επιδράσεις του ιονισμού. [6].

Όσο πιο λεπτό είναι το οξείδιο της πύλης, τόσο πιο λίγα φορτία θα παγιδεύονται σε αυτό και έτσι τόσο πιο μικρή θα είναι η μετατόπιση της τάσης κατωφλίου. Στο σχήμα 1.12 φαίνεται η μετατόπιση της τάσης κατωφλίου ανάλογα με την σμίκρυνση της τεχνολογίας και την ελάττωση του πάχους του οξειδίου. Βλέπουμε ότι όσο μικραίνει το πάχος του οξειδίου, η τάση κατωφλίου αλλάζει μόνο μερικά μV/Mrad σε αντίθεση με τις παλαιές τεχνολογίες όπου η αλλαγή ήταν της τάξης των μερικών δεκάδων ή και εκατοντάδων mV/Mrad.



Σχήμα 1.12: Επίδραση της συνολικής δόσης στην τάση κατωφλίου ενός τρανζίστορ συναρτήσει του πάχους οξειδίου. [6]

Μια μέθοδος η οποία αποσκοπεί στην ελάττωση του παρασιτικού ρεύματος και βασίζεται στον σχεδιασμό του ίδιου του τρανζίστορ, είναι αυτή των κυκλικών τρανζίστορ [22], [23], [24]. Η βασική ιδέα είναι το παρασιτικό τρανζίστορ να έχει την πηγή και τον απαγωγό στο ίδιο δυναμικό αποκλείοντας έτσι την δυνατότητα ύπαρξης παρασιτικού ρεύματος. Μια τέτοια διάταξη φαίνεται στο δεξί μέρος του σχήματος 1.13. Η πηγή περικλείει την πύλη, η οποία με την σειρά της περικλείει τον απαγωγό. Το παρασιτικό τρανζίστορ το οποίο δημιουργείται με αυτή τη γεωμετρία δεν μπορεί να άγει ρεύμα σε αντίθεση με αυτό το οποίο βασίζεται στην κανονική γεωμετρία και το οποίο φαίνεται στο αριστερό μέρος του σχήματος 1.13.



Σχήμα 1.13: Κάτοψη της κανονικής και της κυκλικής γεωμετρίας. [6]

Το πρόβλημα το οποίο έχουνε αυτές οι διατάξεις είναι ότι είναι δύσκολος ο υπολογισμός του λόγου W/L. Στο σχήμα 1.14 φαίνεται η γεωμετρία ενός τέτοιου τρανζίστορ όπως αυτό είναι στην πραγματικότητα.



Σχήμα 1.14: Λεπτομέρεια στην γεωμετρία ενός κυκλικού τρανζίστορ. [6]

Μία άλλη μέθοδος αποσκοπεί στο να εξαφανίσει το ρεύμα διαρροής ανάμεσα σε ένα NMOS και ένα PMOS, τρανζίστορ, τα οποία είναι γειτονικά μεταξύ τους. Αυτό μπορεί να γίνει με την χρήση δακτυλίων προστασίας (guard rings), όπως φαίνεται στο σχήμα 1.15. Τα δύο τρανζίστορς χωρίζονται μεταξύ τους από τους δακτυλίους προστασίας, οι οποίοι εμποδίζουν την διαρροή ρεύματος από το παρασιτικό MOS που σχηματίζεται από το οξείδιο που διαχωρίζει κάθετα το υπόστρωμα τύπου p, στο οποίο βρίσκεται το NMOS, από το πηγάδι τύπου n, στο οποίο βρίσκεται το PMOS [25].



Σχήμα 1.15: Σχεδίαση τρανζίστορς με χρήση δακτυλίων προστασίας. [6]

1.5 Ανθεκτικότητα σε επιδράσεις απλού συμβάντος

Εδώ και χρόνια έχει προβλεφθεί ότι η ευαισθησία στα SEE πρέπει να είναι μεγαλύτερη στις σύγχρονες τεχνολογίες λόγω της μείωσης της τάσης τροφοδοσίας [26] [27].

Η βιβλιογραφία δείχνει ότι τα SETs ενισχύονται με την μείωση της ενέργειας που απαιτείται για τη μετάβαση από τη μία λογικά κατάσταση σε άλλη. Ωστόσο, παρά την παρατηρούμενη αύξηση της πιθανότητας δημιουργίας SET σε περίπλοκα ολοκληρωμένα κυκλώματα με την αύξηση της σμίκρυνσης, η προέκταση πρόβλεψης σε ακόμη πιο νέες τεχνολογίες είναι δύσκολη γιατί υπάρχουν πολλά διαφορετικά φαινόμενα που δρουν σε διαφορετικές κατευθύνσεις [28].

Αυτή η σύντομη ανασκόπηση των επιπτώσεων της μείωσης κλίμακας της τεχνολογίας CMOS στην ευαισθησία των SEE, αποδεικνύει ότι τέτοιες επιδράσεις μπορούν να εξακολουθήσουν να είναι υπεύθυνες για την αποτυχία των κυκλωμάτων σε περιβάλλον με ακτινοβολία και ότι πρέπει να χρησιμοποιηθούν ειδικές τεχνικές για να αποτρέψουν κάτι τέτοιο. Παρακάτω θα δούμε ορισμένες πιθανές λύσεις.

1.5.1 Ανθεκτικότητα σε διαταράξεις απλού συμβάντος (SEU)

Τα στοιχεία μνήμης που χρησιμοποιούνται στους καταχωρητές είναι συνήθως τα περισσότερο ευάλωτα κυκλώματα στα SEUs, επομένως μια πιθανή λύση για να επιτευχθεί ανθεκτικότητα, είναι να τροποποιηθεί η φυσική σχεδίαση των κελιών ή/και η αρχιτεκτονική, ώστε να μειωθεί η ευαισθησία τους. Αυτή η προσέγγιση απαιτεί την κατάλληλη σχεδίαση των κελιών, ώστε να μπορούν να ικανοποιούν τα κριτήρια που ορίζονται για κάθε εφαρμογή, η οποία διαφέρει για διάφορα περιβάλλοντα ακτινοβολίας. Για παράδειγμα, η αύξηση ανθεκτικότητας ενός κυκλώματος για εφαρμογές σε ένα περιβάλλον σωματιδίων που καθορίζεται μόνο από φορτισμένα και ουδέτερα αδρόνια (πρωτόνια, πιόνια, νετρόνια) όπως ένα τυπικό για πειράματα φυσικής υψηλών ενεργειών, είναι πολύ διαφορετική από την αύξηση ανθεκτικότητας για ένα περιβάλλον όπου υπάρχουν βαρέα ιόντα όπως για δορυφορικές εφαρμογές ή αποστολές στο βαθύτερο διάστημα. Η αύξηση της ανθεκτικότητας των κελιών μνήμης, διαφέρει από ότι για τα flip-flops ή για μανταλωτές.

Μια προσέγγιση για την προστασία ενός κυκλώματος από τη SEU, είναι η προσθήκη πλεονασμού στις αποθηκευμένες πληροφορίες. Αυτό μπορεί να γίνει με δύο τρόπους, είτε με τριπλασιασμό των κελιών που αποθηκεύουν τις πληροφορίες (πλεονασμός τριπλών υποσυστημάτων, Triple Modular Redundancy - TMR) ή με την κωδικοποίηση των δεδομένων με τη χρήση τεχνικών ανίχνευσης και διόρθωσης σφαλμάτων (Error Detection And Correction - EDAC).

Όπως αναφέραμε, ο τριπλασιασμός των κελιών αποθήκευσης είναι ένας έγκυρος τρόπος προστασίας του περιεχομένου, αν και υπάρχει σημαντική αύξηση σε επιφάνεια και ισχύ. Μπορεί να χρησιμοποιηθεί ένα κύκλωμα ψηφοφόρου (voter) για να συγκρίνει την έξοδο των τριπλών κελιών μνήμης και στην έξοδό του να προκρίνει τα δεδομένα που υπήρχαν σε πλειοψηφία, αλλά στην περίπτωση αυτή η τελική έξοδος θα μπορούσε να επηρεαστεί από ένα σφάλμα που προκλήθηκε στον ψηφοφόρο. Μια ασφαλέστερη τεχνική είναι να τριπλασιάσουμε τον ψηφοφόρο όπως φαίνεται στο σχήμα 1.16. Σε αυτή τη περίπτωση, η τελική έξοδος προστατεύεται από σφάλματα τόσο στα κελιά μνήμης, όσο και στους ψηφοφόρους. Η προσέγγιση αυτή είναι πολύ απαιτητική σε επιφάνεια, ωστόσο χρησιμοποιείται ευρέως, για παράδειγμα στις εφαρμογές Φυσικής Υψηλών Ενεργειών, όπου είναι συχνά αποδεκτή η αύξηση επιφάνειας.

Σε αντίθεση με την τεχνική TMR όπου κάθε bit πληροφοριών είναι τριπλασιασμένο, η τεχνική EDAC απαιτεί πολύ μικρότερο πλεονασμό πληροφοριών [29]. Το EDAC χρησιμοποιείται πολύ εκτεταμένα για μνήμες ημιαγωγών, αλλά και οπουδήποτε τα δεδομένα πρέπει να αποθηκεύονται αξιόπιστα (για παράδειγμα, σε CD και DVD χρησιμοποιείται εκτεταμένα). Οι πληροφορίες που πρέπει να αποθηκευτούν κωδικοποιούνται από ένα πολύπλοκο λογικό μπλοκ, και μερικές πλεονάζουσες πληροφορίες προστίθενται σε αυτή τη διαδικασία. Όσο μεγαλύτερος είναι ο αριθμός των πλεοναζόντων bit, τόσο πιο ασφαλή είναι τα αρχικά δεδομένα από οποιαδήποτε πηγή σφάλματος και αν προέρχονται. Τη στιγμή της ανάγνωσης των αποθηκευμένων πληροφοριών, για την αποκωδικοποίηση είναι απαραίτητο ένα άλλο πολύπλοκο σύνολο λογικών λειτουργιών. Αρκετοί κώδικες μπορούν να χρησιμοποιηθούν για το EDAC, όπως οι Hamming, Reed-Solomon και BCH [30], και κάθε ένα από αυτά έχει διαφορετικές δυνατότητες ανίχνευσης και διόρθωσης, και διαφορετική πολυπλοκότητα.



Σχήμα 1.16: Αύξηση ανθεκτικότητας σε SEU με πλεονασμό: Το TMR εφαρμόζεται σε όλες τις πληροφορίες καθώς και στους ψηφοφόρους. [26]

Σε επόμενο κεφάλαιο θα γίνει εκτενέστερη αναφορά στο TMR, καθώς θα χρησιμοποιήσουμε τη συγκεκριμένη μέθοδο.

1.5.2 Ανθεκτικότητα σε κλείδωμα απλού συμβάντος (SEL)

Πολλές μέθοδοι έχουν αναπτυχθεί για να γίνουν τα ολοκληρωμένα κυκλώματα ανθεκτικά έναντι του SEL. Οι μέθοδοι για αύξηση της ανθεκτικότητας κατά τη σχεδίαση, όπως η εφαρμογή δακτυλίων προστασίας (guard rings) [31], δεν απαιτούν καμία τροποποιήση στην διεργασία κατασκευής των ημιαγωγών, αλλά αυξάνουν την επιφάνεια που καταλαμβάνει ένα ολοκληρωμένο κύκλωμα. Η αύξηση της ανθεκτικότητας κατά την διεργασία, όπως είναι οι μέθοδοι επεξεργασίας που χρησιμοποιούν ένα επιταξιακό στρώμα πάνω σε ένα υπόστρωμα βαριά νοθευμένο, δεν έχει καμία επίπτωση στην επιφάνεια, αλλά απαιτεί τροποποιήσεις στη διεργασία. Όλες οι μέθοδοι αύξησης της ανθεκτικότητας κατά τρονζίστορ, μειώνοντας το κέρδος τους, ή μειώνοντας τις αντιστάσεις στις επαφές πηγαδιού-υποστρώματος (όπως είχαμε αναφέρει και παραπάνω, σχήμα 1.1). Η πιο κατάλληλη μέθοδος αύξησης της ανθεκτικότητας σε SEL για μια εφαρμογή εξαρτάται από πολλούς παράγοντες, συμπεριλαμβανομένων της αύξησης στο μέγεθος της επιφάνειας που μπορεί να γίνει ανεκτή και το απαιτούμενο επίπεδο αντοχής.
Το κλείδωμα εμφανίζεται μόνο όταν δημιουργούνται επαρκώς μεγάλες πτώσεις τάσης στις αντιστάσεις του υποστρώματος και του Ν-τύπου πηγαδιού, οι οποίες πολώνουν ορθά τις επαφές εκπομπού-βάσης των pnp και npn τρανζίστορ αντίστοιχα. Επομένως, κάθε αλλαγή που μειώνει αυτές τις παρασιτικές αντιστάσεις, κάνει πιο δύσκολο να ενεργοποιηθούν τα διπολικά τρανζίστορ, πράγμα που αυξάνει την ανθεκτικότητα του κλειδώματος.

Οι δακτύλιοι προστασίας είναι μία αποτελεσματική μέθοδος για την αποσύζευξη των BJT, γιατί μειώνουν τον κίνδυνο αλληλεπιδράσεων μεταξύ των διατάξεων, παρέχοντας ηλεκτρική και χωρική απομόνωση [32]. Ένας δακτύλιος προστασίας (n+ guard ring) είναι ουσιαστικά ένα πηγάδι που εκτείνεται κατά μήκος της περιμέτρου του Ν-πηγαδιού, σε Ρυπόστρωμα, και περιβάλλει όλες τις διατάξεις PMOS (σχήμα 1.17). Αυτό βοηθά στην αποφυγή των πτώσεων τάσης που μπορεί να οδηγήσουν σε κλείδωμα.



Σχήμα 1.17: CMOS τρανζίστορς με δακτύλιο προστασίας. [32]

2. ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΤΩΝ ΚΕΛΙΩΝ

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δούμε τι είναι ο χαρακτηρισμός και τι πληροφορίες παίρνουμε για το κάθε κελί, όπου κελί είναι κάποια πύλη, ένα flip-flop, ένας αθροιστής κλπ., θα δούμε τι είναι τα ακρότατα σημεία και τα μοντέλα χρονισμού και ισχύος που χρησιμοποιούνται στον χαρακτηρισμό και πως παρουσιάζονται στο πρότυπο Liberty που είναι το αρχείο (.lib) που περιλαμβάνει όλες τις πληροφορίες του χαρακτηρισμού. Επίσης θα γίνει αναφορά στα κελιά που θα χαρακτηριστούν και στη διαδικασία χαρακτηρισμού και τέλος θα γίνει σύγκριση των αποτελεσμάτων με τα πρωτότυπα κελιά.

Ο χαρακτηρισμός είναι μια διαδικασία ανάλυσης για τον προσδιορισμό των ηλεκτρικών χαρακτηριστικών (χρονισμοί, ισχύς) ενός κελιού. Αυτά τα χαρακτηριστικά είναι απαραίτητα για τη ροή της σχεδίασης ενός ψηφιακού κυκλώματος για την περαιτέρω ανάλυση του (κατανάλωση ισχύος, ανάλυση χρονισμού, προσομοίωση). Όλες αυτές οι πληροφορίες αποθηκεύονται σε ένα αρχείο, το οποίο περιέχει τα χαρακτηριστικά για ένα σύνολο κελιών. Αυτό το αρχείο είναι η βιβλιοθήκη για το πρόγραμμα σχεδίασης και χρησιμοποιείται σε όλα τα στάδια της σχεδίασης, από τη λογική σχεδίαση μέχρι την τελική επαλήθευση. Η κάθε βιβλιοθήκη περιέχει τα χαρακτηριστικά των κελιών για συγκεκριμένες συνθήκες, όπως τάση λειτουργίας, θερμοκρασία και ακρότατο σημείο στην διακύμανση των παραμέτρων της διεργασίας κατασκευής (process corner).

2.2 Ακρότατα σημεία διεργασίας κατασκευής ολοκληρωμένων κυκλωμάτων

Ιδανικά σε κάθε δισκίδιο πυριτίου και σε όλη την επιφάνεια του, ίδια κυκλώματα έχουν τα ίδια ακριβώς χαρακτηριστικά. Αλλά αυτό δεν είναι εφικτό, γιατί αφενός δεν μπορούν να ελεγχθούν όλοι οι παράγοντες με απόλυτη επαναληψιμότητα και αφετέρου οι διεργασίες που υφίσταται το πυρίτιο για την κατασκευή των ολοκληρωμένων κυκλωμάτων έχουν από τη φύση τους μία συνιστώσα τυχαιότητας. Μερικές από τις παραμέτρους ενός τρανζίστορ, όπως το μήκος του καναλιού, η τάση κατωφλίου κ.α. μπορεί να ποικίλουν σε κάποιο βαθμό, γι αυτό και η συμπεριφορά των τρανζίστορ ποικίλει αναλόγως.

Σε κάθε δισκίδιο πυριτίου υπάρχουν δομές ελέγχου που μετριούνται κατά την διαδικασία της κατασκευής και αν οι τιμές τους είναι έξω από κάποια όρια το δισκίδιο απορρίπτεται. Αυτά τα εγγυημένα όρια είναι που καθορίζουν τα ακρότατα σημεία (process corners) που λαμβάνει υπόψη του ο σχεδιαστής. Αυτά τα όρια χωρίζονται σε τρεις κατηγορίες. Για καλύτερες συνθήκες (best), για χειρότερες (worst) και για τυπικές (typical). Συνήθως τις συναντάμε με μια ορολογία δύο γραμμάτων SS, TT, FF, FS και SF, όπου αυτά σημαίνουν Slow-Slow (αργό-αργό), Typical-Typical (τυπικό-τυπικό), Fast-Fast (γρήγορο-γρήγορο), Fast-Slow (γρήγορο- αργό) και Slow-Fast (αργό γρήγορο). Το πρώτο γράμμα αντιστοιχεί στο NMOS και το δεύτερο στο PMOS τρανζίστορ ενός CMOS κυκλώματος. Το ακρότατο TT ουσιαστικά δεν υπάρχει, καθώς είναι οι ονομαστικές τιμές των χαρακτηριστικών των τρανζίστορ [33].

Στα σημεία SS, TT και FF μπορούμε να αναμένουμε ισχυρή συσχέτιση στη συμπεριφορά PMOS και NMOS, καθώς και τα δύο θα είναι αργά ή γρηγορότερα. Οι γωνίες SF και FS στις οποίες η καθυστέρηση που παρουσιάζουν τα PMOS και NMOS είναι διαφορετική, επομένως το σήμα εξόδου έχει διαφορετικούς χρόνους μεταβάσεων στις αιχμές ανόδου απ' ότι στις αιχμές καθόδου. Το PMOS ελέγχει τις μεταβάσεις στις ανόδους και το NMOS στις καθόδους του σήματος εξόδου. Συνήθως δεν θεωρούμε ακρότατα σημεία τα SF και FS επειδή καλύπτονται στα σημεία SS και FF. Επομένως για να διασφαλίσουμε την σωστή λειτουργία ενός κυκλώματος πρέπει πάντα να το δοκιμάζουμε στις ακραίες

περιπτώσεις, δηλαδή στα σημεία SS και FF. Στο σχήμα 2.1 απεικονίζεται τα ακρότατα σημεία.



Σχήμα 2.1: Ακρότατα σημεία. [34]

Τα χαρακτηριστικά ενός τρανζίστορ ποικίλλουν επίσης λόγω της τάσης και της θερμοκρασίας λειτουργίας. Οι διατάξεις ημιαγωγών λειτουργούν καλύτερα με μεγαλύτερη τάση και σε χαμηλότερες θερμοκρασίες. Για να αποκτήσουμε τα πραγματικά ακρότατα σημεία, θα πρέπει να τις συνδέσουμε με την τάση και τη θερμοκρασία λειτουργίας.

2.3 Ροή χαρακτηρισμού

Χαρακτηρισμός είναι η διαδικασία της εξαντλητικής ανάλυσης μιας ομάδας κελιών για την εξαγωγή όλων των σχετικών και ουσιαστικών πληροφοριών γι' αυτά. Ο χαρακτηρισμός των κελιών είναι το θεμέλιο στο οποίο βασίζεται όλη η ψηφιακή σχεδίαση. Στο σχήμα 2.2 παρουσιάζεται το διάγραμμα ροής του χαρακτηρισμού.



Σχήμα 2.2: Διάγραμμα ροής χαρακτηρισμού. [35]

Όπως βλέπουμε και από το παραπάνω σχήμα, το εργαλείο χαρακτηρισμού για να πραγματοποιήσει τον χαρακτηρισμό των κελιών, πρέπει να εισάγουμε δεδομένα. Αυτά είναι τα εξής [35]:

Περιγραφή δικτυώματος των κελιών και των παρασιτικών στοιχείων τους σε μορφή SPICE (SPICE parasitic netlist). Τα τρανζίστορ, οι δίοδοι, οι αντιστάσεις, οι πυκνωτές και τα εξαγόμενα παρασιτικά στοιχεία (RC) που απαρτίζουν ένα κελί, περνάνε στο εργαλείο σε μορφή SPICE. Τα εξαγόμενα αρχεία με την περιγραφή δικτυώματος (netlists) μπορούν να εξαχθούν απευθείας, μέσω των εργαλείων, από τη φυσική σχεδίαση (layout) του κελιού. Τα πρότυπα που υποστηρίζονται για τα netlist αρχεία, είναι τα SPICE και Hspice καθώς και το Spectre. Τα netlist των κελιών που θα χαρακτηριστούν μπορούν να είναι σε ένα και μόνο αρχείο ή σε ομάδα αρχείων.

Μοντέλα των συσκευών. Τα μοντέλα των συσκευών παρέχονται από το εργοστάσιο κατασκευής και αντιπροσωπεύουν τις ηλεκτρικές παραμέτρους τους για κάθε ακρότατο σημείο (SS, TT, FF, FS και SF). Αυτά τα αρχεία περιλαμβάνουν μοντέλα από τρανζίστορ (κανάλι P και N), διόδους, πυκνωτές και αντιστάσεις.

Αρχείο με το σχεδιότυπο (template). Αυτό είναι ένα αρχείο που δημιουργείται από τον χρήστη και είναι γραμμένο σε γλώσσα προγραμματισμού σεναρίων tcl. Αυτό το αρχείο περιλαμβάνει διάφορες ρυθμίσεις που αφορούν τον χαρακτηρισμό και τι είδους πληροφορίες θα εξαχθούν για το κάθε κελί, όπως το εύρος των δεδομένων για το οποίο θα γίνει ο χαρακτηρισμός, π.χ. τον χρόνο ανόδου της εισόδου (input slew) και χωρητικότητα εξόδου, καθώς και για ποιόν συνδυασμό διεγέρσεων στις εισόδους του κάθε κελιού θα εξεταστεί η κατανάλωση ισχύος κλπ.

Για να ελέγξουμε τη διαδικασία χαρακτηρισμού, χρησιμοποιούμαι επίσης τη γλώσσα tcl. Αρχικά καθορίζουμε τα netlists των κελιών, τα μοντέλα SPICE και τις συνθήκες λειτουργίας. Το εργαλείο χαρακτηρισμού θα εκτελέσει τις προσομοιώσεις και θα κάνει μετρήσεις στα αποτελέσματα τους για κάθε κελί χρησιμοποιώντας τα δεδομένα που του έχουμε ορίσει από το template και θα δημιουργήσει τους κατάλληλους πίνακες με πληροφορίες ισχύος (ισχύς μετάβασης, κρυφή ισχύς, ισχύς διαρροής), χρόνων προετοιμασίας (setup), συγκράτησης (hold) κ.λπ.

Τέλος, το εργαλείο χαρακτηρισμού γράφει όλους αυτούς τους πίνακες σε ένα .lib αρχείο [36], το οποίο περιέχει όλες τις πληροφορίες για το κάθε κελί. Κάθε αρχείο αντιστοιχεί σε μία σημείο διεργασίας, μία θερμοκρασία και μία τάση λειτουργίας. Για περισσότερες συνθήκες λειτουργίας, θα πρέπει να γίνει ξεχωριστά χαρακτηρισμός για την κάθε μία.

Από τον χαρακτηρισμό μπορούμε να πάρουμε τις εξής πληροφορίες για το κάθε κελί:

- Καθυστέρηση (delay). Περιλαμβάνει τον χρόνο μετάβασης και την καθυστέρηση διάδοσης.
- Χρονικοί περιορισμοί (constraints). Περιλαμβάνουν τον χρόνο προετοιμασίας (set-up time), τον χρόνο συγκράτησης (hold time), τον χρόνο αποκατάστασης (recovery time) και τον χρόνο απομάκρυνσης (removal time).
- Την κατανάλωση στατικής ισχύος (static power). Περιλαμβάνει την ισχύ διαρροής (leakage power).
- Την κατανάλωση δυναμικής ισχύος (dynamic power). Περιλαμβάνει την ισχύ μεταγωγής (transition power) και την κρυφή ισχύ (hidden power).

Χρόνος μετάβασης (Transition time), είναι ο χρόνος που χρειάζεται ένα σήμα να μεταβεί από μία κατάσταση σε άλλη. Το σχήμα που ακολουθεί δείχνει ένα παράδειγμα που ο χρόνος μετάβασης είναι για την άνοδο από το 10% έως το 90% (χρόνος ανόδου - rise

time) και για την καθόδου από το 90% έως το 10% (χρόνος καθόδου - fall time). Αυτά τα όρια ορίζονται από τον χρήστη.



Σχήμα 2.3: Χρόνος μετάβασης ενός σήματος. [37]

Καθυστέρηση διάδοσης (Propagation delay). Είναι ο χρόνος που απαιτείται για να γίνει μετάβαση στο σήμα εξόδου, μετά από αλλαγή στο σήμα εισόδου. Κανονικά, ορίζεται ως η διαφορά μεταξύ των χρόνων κατά τις οποίες η μετάβαση της εισόδου φθάνει το 50% της τελικής της τιμής, μέχρι τη στιγμή που η έξοδος φτάνει στο 50% της τελικής τιμής. Εδώ, το 50% ορίζεται ως το λογικό όριο όπου η έξοδος (ή οποιοδήποτε σήμα) υποτίθεται ότι αλλάζει την κατάσταση του. Στο σχήμα 2.4 απεικονίζεται η καθυστέρηση διάδοσης μιας πύλης AND. Υπάρχει μεταβολή της εισόδου Ι2 και με καθυστέρηση Τρ μεταβάλλεται η έξοδος.



Σχήμα 2.4: Καθυστέρηση διάδοσης μίας πύλης AND. [38]

Χρόνος προετοιμασίας (setup time) είναι ο ελάχιστος χρόνος που απαιτείται να έχουν φτάσει τα δεδομένα στην εισόδου ενός ακολουθιακού στοιχείου (flip-flop) πριν την έλευση της ενεργής αιχμής του ρολογιού.

Χρόνος συγκράτησης (hold time) αντίστοιχα είναι ο ελάχιστος χρόνος που απαιτείται να παραμείνουν τα δεδομένα στην εισόδου ενός ακολουθιακού στοιχείου (flip-flop) μετά την έλευση της ενεργής αιχμής του ρολογιού. Το σχήμα 2.5 μας βοηθάει να καταλάβουμε την έννοια των χρόνων προετοιμασίας και συγκράτησης. Στον κόκκινο κύκλο βλέπουμε το σήμα στην είσοδο ενός flip-flop να έχει μικρότερη διάρκεια από τον απαιτούμενο χρόνο και αυτό μπορεί να έχει σαν συνέπεια την εγγραφή λανθασμένης τιμής στο flip flop.



Σχήμα 2.5: Κατανόηση των χρόνων προετοιμασίας και συγκράτησης. [37]

Χρόνος αποκατάστασης (recovery time) είναι ο ελάχιστος χρόνος που απαιτείται για την διατήρηση μίας ασύγχρονης εισόδου (set/clear) ενός flip-flop, πριν την έλευση της ενεργής αιχμής του ρολογιού.

Χρόνος απομάκρυνσης (removal time) είναι ο ελάχιστος χρόνος που απαιτείται για την διατήρηση μίας ασύγχρονης εισόδου (set/clear) ενός flip-flop, μετά την έλευση της ενεργής αιχμής του ρολογιού. Στο σχήμα 2.6 απεικονίζεται με τις μπλε διακεκομμένες ο χρόνος που πρέπει να ενεργοποιηθεί το reset πριν την έλευση του ρολογιού και με τις κόκκινες διακεκομμένες ο χρόνος που πρέπει να περάσει από την έλευση του ρολογιού για την ενεργοποίηση του reset.



Σχήμα 2.6: Χρόνοι αποκατάστασης και απομάκρυνσης ενός flip-flop. [37]

Στατική ισχύς (Static power) ή αλλιώς **leakage power** είναι η ισχύς που καταναλώνει ένα κελί όταν είναι σε σταθερή κατάσταση (κατάσταση αναμονής) και δεν υπάρχει μετάβαση σε κάποια είσοδο ή έξοδο και υπολογίζεται για κάθε δυνατό συνδυασμό εισόδων.

Ισχύς μεταγωγής (Switching power) είναι η ισχύς που καταναλώνεται από το κελί όταν μία αλλαγή σε μία είσοδο προκαλέσει την αλλαγή μίας εξόδου.

Κρυφή ισχύς (Hidden power) είναι η ισχύς που καταναλώνεται μέσα στο κελί όταν υπάρχει αλλαγή σε μια είσοδο και δεν οδηγεί σε κάποια αλλαγή την έξοδο.

2.4 Μοντελοποίηση [39]

Η μοντελοποίηση είναι ένας τρόπος να συνοψίσουμε τα βασικά χαρακτηριστικά ενός κυκλώματος (όπως ενέργεια, χρονισμοί κλπ) σε ένα κελί. Αυτά τα μοντέλα είναι απαραίτητα για τη σχεδίαση κυκλωμάτων σε ολοκληρωμένα κυκλώματα ειδικών εφαρμογών (Application Specific Integrated Circuit - ASIC).

Η προσομοίωση των κυκλωμάτων είναι μια διαδικασία εντατικών υπολογισμών. Όσο η πολυπλοκότητα των κυκλωμάτων αυξάνει και ένα ολοκληρωμένο κύκλωμα μπορεί να έχει δισεκατομμύρια τρανζίστορ, είναι αδύνατο να γίνει προσομοίωση σε σύντομο χρονικό διάστημα. Αρκετές μεθοδολογίες σχεδίασης βασισμένες σε κελιά εξελίχθηκαν για την αντιμετώπιση αυτού το προβλήματος και την ταχύτερη σχεδίαση των ολοκληρωμένων κυκλωμάτων. Αυτές οι μεθοδολογίες σχεδίασης (ASIC, FPGA κλπ) βασίζονται στη λεπτομερή μοντελοποίηση του υποκείμενου κυκλώματος ως ένα κελί. Οι πληροφορίες στα κελιά διευκολύνουν το λογισμικό σχεδίασης (όπως σύνθεση, ανάλυση χρονισμού και ισχύος κλπ) να εκτελεί αυτούς τους υπολογισμούς κατά πολλές τάξεις μεγέθους ταχύτερα από ότι η αναλογική προσομοίωση του κυκλώματος.

Υπάρχουν πολλά χαρακτηριστικά σε μία VLSI σχεδίαση (ισχύς, χρονισμός, θόρυβος, φυσική επιφάνεια κ.α.). Έχουν δημιουργηθεί πολλά πρότυπα που αναφέρονται στην διαχείριση καθενός από αυτά τα χαρακτηριστικά. Η ανάλυση χρονισμού, ισχύος και ο θόρυβος καλύπτονται από το πρότυπο Liberty, προερχόμενο από την Synopsys. Οι φυσικές πληροφορίες καλύπτονται από τα πρότυπα LEF (Layout Exchange Format) και DEF (Design Exchange Format). Αυτά τα αρχεία περιλαμβάνουν τους κανόνες σχεδίασης για την διασύνδεση των κελιών, τις διαστάσεις του κελιού, τις αντιστάσεις και τις χωρητικότητες των μετάλλων κ.α.

Το πρότυπο Liberty είναι ανοιχτού κώδικα και έχει ευρεία υποστήριξη από τη βιομηχανία ημιαγωγών, με πάνω από 100 κατασκευαστές να το χρησιμοποιούν. Το Liberty έχει διαφορετικούς τύπους μοντέλων για τον χρόνο, την ενέργεια και τον θόρυβο. Αυτά είναι τα εξής:

- NLDM (Non-Linear Delay Model)
- NLPM (Non-Linear Power Model)
- ECSM (Effective Current Source Model)
- CCS (Composite Current Source)
- CCSP (CCS Power)
- CCSN (CCS Noise)

Τα μοντέλα ECSM, CCS, CCSP, CCSN είναι ακριβέστερα, αλλά απαιτούν περισσότερο χρόνο και έχουν μεγαλύτερο μέγεθος αρχείων και είναι κατάλληλα για τεχνολογίες μικρότερες των 130 nm. Στη παρούσα εργασία θα χρησιμοποιήσουμε τα NLDM και NLPM, επομένως θα γίνει αναφορά μόνο σε αυτά.

2.4.1 Μοντέλο χρονισμού

Πριν σταλεί ένα σχέδιο στο εργοστάσιο κατασκευής για την υλοποίηση του ολοκληρωμένου κυκλώματος πραγματοποιείται στατική ανάλυση χρονισμού (STA - Static Timing Analysis), ώστε να εξασφαλιστεί η χρονικά σωστή λειτουργία του κυκλώματος. Η στατική ανάλυση χρονισμού πραγματοποιείται με μία συλλογή μοντέλων χρονισμού για κάθε κελί. Το μοντέλο χρονισμού είναι ένα σετ που περιλαμβάνει τα τόξα καθυστέρησης, τόξα χρονικού περιορισμού (χρόνος προετοιμασίας, χρόνος συγκράτησης, χρόνος αποκατάστασης και χρόνος απομάκρυνσης) και τις χωρητικότητες των εισόδων για το κάθε κελί.

2.4.2 Χρονικά τόξα (Timing arcs)

Το χρονικό τόξο είναι μια έννοια της χρονικής εξάρτησης μεταξύ των σημάτων σε οποιαδήποτε δύο σχετιζόμενους ακροδέκτες ενός κελιού. Το παρακάτω σχήμα δείχνει μία απλοποιημένη μορφή των χρονικών τόξων με βέλη. Μία πύλη AND σημειώνεται με δύο χρονικά τόξα. Από την είσοδο Χ προς την έξοδο Υ και από την είσοδο Β στην έξοδο Υ.



Σχήμα 2.7: Χρονικά τόξα σε μία πύλη AND. [39]

Τα χρονικά τόξα ταξινομούνται με βάση την κατεύθυνση του σήματος των ακροδεκτών του κελιού. Ένας ακροδέκτης μπορεί να έχει δύο κατευθύνσεις, την είσοδο και την έξοδο. Αυτές οι κατευθύνσεις επιτρέπουν τα παρακάτω τόξα:

- Τόξα καθυστέρησης (από είσοδο προς έξοδο)
- Τόξα χρονικού περιορισμού (από είσοδο προς είσοδο).

2.4.2.1 Τόξα καθυστέρησης

Με τα τόξα καθυστέρησης μετράμε την καθυστέρηση διάδοσης και τον χρόνο μετάβασης. Με βάση τα παραπάνω, χρειάζονται δύο ακροδέκτες για να σχηματιστεί ένα τόξο καθυστέρησης. Ο πρώτος είναι ένας ακροδέκτης εξόδου και ο δεύτερος είναι ένας σχετιζόμενος με τον ακροδέκτη εξόδου. Ο δεύτερος ακροδέκτης μπορεί να είναι μία είσοδος ή μία έξοδος.

Το NLDM μοντέλο υπολογίζει την καθυστέρηση διάδοσης από τον σχετιζόμενο ακροδέκτη εισόδου προς στον ακροδέκτη εξόδου και το χρόνο μετάβασης στον ακροδέκτη εξόδου για κάθε μετάβαση. Τα ακόλουθα τέσσερα γκρουπ καλύπτουν τον τύπο της μετάβασης, την καθυστέρηση και τον χρόνο μετάβασης:

- "Cell_rise" την καθυστέρηση διάδοσης για την ανόδου του σήματος στον ακροδέκτη εξόδου.
- 2. "Cell_fall" την καθυστέρηση διάδοσης για την κάθοδο του σήματος στον ακροδέκτη εξόδου.
- "Rise_transition" τον χρόνο μετάβασης για την άνοδο του σήματος στον ακροδέκτη εξόδου.
- 4. "Fall_transition" τον χρόνο μετάβασης για την κάθοδο του σήματος στον ακροδέκτη εξόδου.

	"Cell_ri	se"			"Cell_fa	all"	
Index1/Index2	C1	C2	C3	Index1/Index2	C1	C2	C3
S1	Ds1C1	Ds1c2	Ds1c3	S1	Ds1C1	Ds1c2	Ds1c3
S2	Ds2C1	Ds2c2	Ds2c3	S2	Ds2C1	Ds2C2	Ds2C3
S3	Ds3C1	Ds3c2	Ds3c3	S3	Ds3C1	Ds3c2	Ds3c3
	'rise_trans	sition"			'fall_trans	ition"	
Index1/Index2	C1	C2	C3	Index1/Index2	C1	C2	C3
S1	Ts1c1	Ts1C2	T s1c3	S1	Ts1C1	T s1c2	Ts1c3
S2	T s2C1	Ts2c2	T _{S2C3}	S2	T _{S2C1}	Ts2c2	Ts2c3
S3	T s3C1	T _{S3C2}	Тзэсэ	S3	T _{S3C1}	Ts3c2	Тззсз

Σχήμα 2.8: NLDΜ τόξα καθυστέρησης.

Αυτά τα γκρουπ τοποθετούνται στο γκρουπ "timing" στο πρότυπο Liberty και απεικονίζονται στον αντίστοιχο πίνακα για διαφορετικούς συνδυασμούς μεταβάσεων εισόδου (index 1) και χωρητικοτήτων εξόδου (index 2), όπως δείχνουν οι πίνακες του παραπάνω σχήματος 2.8. Όπου 'S', οι διαφορετικές μεταβάσεις εισόδου (input slew), όπου 'C' οι διαφορετικές χωρητικότητες εξόδου, όπου 'D' η καθυστέρηση για το ζευγάρι 'S' και 'C', και όπου 'T' ο χρόνος μετάβασης για το ζευγάρι 'S' και 'C.

2.4.2.2 Συνδυαστικά τόξα καθυστέρησης

Τα τόξα που είναι μέρος των συνδυαστικών κελιών, λέγονται συνδυαστικά τόξα. Ένα σημαντικό μέρος ενός τόξου καθυστέρησης είναι οι σχέσεις στο ζευγάρι των ακροδεκτών,

δηλαδή ο τρόπος με τον οποίο ο σχετιζόμενος ακροδέκτης επηρεάζει έναν ακροδέκτη εξόδου. Όταν η μετάβαση στον σχετιζόμενο ακροδέκτη προκαλεί μετάβαση σε μία η περισσότερες εξόδους, τότε λέμε ότι έχουν μονόσημη σχέση (unate). Υπάρχουν τρείς διαφορετικοί τύποι αυτών των σχέσεων.

- 1. Θετική μονόσημη.
- 2. Αρνητική μονόσημη.
- 3. Μη μονόσημη.

Ένα ζευγάρι ακροδεκτών λέμε ότι έχει θετική μονόσημη σχέση όταν η μετάβαση του σχετιζόμενου ακροδέκτη προκαλεί την ίδια μετάβαση στην έξοδο. Για παράδειγμα μια ανοδική μετάβαση στην είσοδο μίας πύλης AND προκαλεί ανοδική μετάβαση στην έξοδο της πύλης, όταν οι υπόλοιπες είσοδοι είναι στη λογική κατάσταση '1'. Όμοια λέμε ότι έχει αρνητική μονόσημη σχέση όταν η μετάβαση του σχετιζόμενου ακροδέκτη προκαλεί την αντίθετη μετάβαση στην έξοδο. Για παράδειγμα μια στη λογική κατάσταση '1'. Όμοια λέμε ότι έχει αρνητική μονόσημη σχέση όταν η μετάβαση του σχετιζόμενου ακροδέκτη προκαλεί την αντίθετη μετάβαση στην έξοδο. Για παράδειγμα όταν σε έναν αντιστροφέα (πύλη NOT) η είσοδος έχει ανοδική μετάβαση, στην έξοδο προκαλεί καθοδική μετάβαση. Τέλος λέμε ότι έχει μη μονόσημη σχέση όταν η μετάβαση του σχετιζόμενου ακροδέκτη μπορεί να προκαλέσει είτε ανοδική είτε καθοδική μετάβαση στην έξοδο. Για παράδειγμα όταν σε έναν αντιστροφέα (πύλη NOT) η είσοδος έχει ανοδική είτε καθοδική μετάβαση του σχετιζόμενου ακροδέκτη μπορεί να προκαλέσει είτε ανοδική είτε καθοδική μετάβαση στην έξοδο. Για παράδειγμα μία ανοδική μετάβαση στην έξοδο της πύλης ΑΟR δύο εισόδων, μπορεί να προκαλέσει είτε ανοδική είτε καθοδική μετάβαση στην έξοδο ανάλογα με το αν η άλλη είσοδος είναι στην λογική κατάσταση '0' ή στην λογική κατάσταση '1' αντίστοιχα.

Τα τόξα καθυστέρησης βρίσκονται στο γκρουπ "pin" (ακροδέκτης) με κατεύθυνση "output" στο .lib αρχείο. Κάθε γκρουπ ακροδέκτη εξόδου περιέχει ένα γκρουπ "timing" μέσα του. Αυτό το γκρουπ πρέπει να περιέχει τον σχετιζόμενο ακροδέκτη με την κωδική λέξη "related_pin" και τη σχέση μεταξύ των δύο ακροδεκτών με την κωδική λέξη "timing_sense" και προσδιορίζουν όλους τους πιθανούς συνδυασμούς για τα συνδυαστικά κελιά. Η τιμή που έχει το "related_pin" είναι η τιμή που έχει το "timing_sense" περιγράφει την μονόσημη σχέση μεταξύ των δύο ακροδεκτών.

Οι μη μονόσημες σχέσεις αυξάνουν την πολυπλοκότητα της μοντελοποίησης από τη στιγμή που απαιτούν προσδιορισμό της λογικής κατάστασης των υπόλοιπων εισόδων. Για παράδειγμα η έξοδος μίας πύλης XOR δύο εισόδων εξαρτάται από τη λογική κατάσταση της μη σχετιζόμενης εισόδου. Έτσι κάθε "timing" γκρουπ επαναλαμβάνεται δύο φορές για κάθε πιθανή κατάσταση της μη σχετιζόμενης εισόδου, επομένως θα έχει τέσσερα "timing" γκρουπ. Ο αριθμός των "timing" γκρουπ ανεβαίνει εκθετικά για κάθε μία επιπλέον είσοδο.

2.4.2.3 Ακολουθιακά τόξα καθυστέρησης

Ακολουθιακά κελιά ορίζονται τα κελιά που των οποίων η έξοδος δεν εξαρτάται μόνο από την παρούσα κατάσταση της εισόδου, αλλά και από την προηγούμενη κατάσταση. Με άλλα λόγια τα ακολουθιακά κελιά θυμούνται την τελευταία κατάσταση τους. Αυτά τα κελιά έχουν δύο τύπους τόξων καθυστέρησης, τα σύγχρονα τόξα καθυστέρησης και τα ασύγχρονα τόξα καθυστέρησης. Τα σύγχρονα τόξα έχουν σαν σχετιζόμενο ακροδέκτη το ρολόι (clock) του κελιού, ενώ τα ασύγχρονα έχουν σχετιζόμενο ακροδέκτη κάποια ασύγχρονη είσοδο (set, reset).

Τα ακολουθιακά τόξα καθυστέρησης βρίσκονται στο γκρουπ "pin" με κατεύθυνση "output" στο .lib αρχείο. Κάθε γκρουπ ακροδέκτη εξόδου περιέχει ένα γκρουπ "timing" μέσα του. Αυτό το γκρουπ περιέχει τον σχετιζόμενο ακροδέκτη με την λέξη κλειδί "related_pin", τη σχέση μεταξύ των δύο ακροδεκτών με την λέξη κλειδί "timing_sense" και τον τύπο του χρονισμού με την λέξη κλειδί "timing_type". Η τιμή που έχει το "related_pin" είναι το όνομα

του ακροδέκτη που προκαλεί μετάβαση στον ακροδέκτη της εξόδου και η τιμή που έχει το "timing_sense" περιγράφει την μονόσημη σχέση μεταξύ των δύο ακροδεκτών. Η αναγνώριση των σύγχρονων και ασύγχρονων σημάτων παρέχεται από το πεδίο "timing_type", το οποίο δείχνει σε ποια αιχμή του ρολογιού δουλεύει ο αναφερόμενος σύγχρονος ακροδέκτης ή για τους ασύγχρονους ακροδέκτες, τι τύπος είναι (set ή reset). Το "timing_type" μπορεί να έχει τις παρακάτω τιμές:

- Rising_edge: Δείχνει ότι είναι σύγχρονο τόξο και η αναφερόμενη έξοδος είναι ευαίσθητη στην ανοδική αιχμή του σχετιζόμενου ακροδέκτη.
- Falling_edge: Δείχνει ότι είναι σύγχρονο τόξο και η αναφερόμενη έξοδος είναι ευαίσθητη στην καθοδική αιχμή του σχετιζόμενου ακροδέκτη.
- Preset: Δείχνει ότι είναι ασύγχρονο τόξο και η αναφερόμενη έξοδος μεταβαίνει στην λογική κατάσταση '1', όταν ο σχετιζόμενος ακροδέκτης (set ή preset) ενεργοποιηθεί. Ανάλογα με το κελί, η ενεργοποίηση του ακροδέκτη μπορεί να γίνεται με '0' ή με '1'.
- Clear: Δείχνει ότι είναι ασύγχρονο τόξο και η αναφερόμενη έξοδος μεταβαίνει στην λογική κατάσταση '0', όταν ο σχετιζόμενος ακροδέκτης (clear ή reset) ενεργοποιηθεί. Ανάλογα με το κελί, η ενεργοποίηση του ακροδέκτη μπορεί να γίνεται με '0' ή με '1'.

2.4.3 Τόξα χρονικού περιορισμού (constraint arcs)

Ο χρονικός περιορισμός είναι η χρονική διαφορά μεταξύ δύο σημάτων που πρέπει να τηρηθεί όλες τις φορές για να λειτουργήσει σωστά το κύκλωμα. Τα τόξα χρονικού περιορισμού δημιουργούνται μεταξύ δύο σημάτων. Όταν το σήμα αναφοράς είναι το ρολόι, ονομάζεται ακολουθιακό τόξο χρονικού περιορισμού. Όταν και τα δύο σήματα είναι σήματα δεδομένων, τότε ονομάζεται μη ακολουθιακό τόξο χρονικού περιορισμού.

2.4.3.1 Ακολουθιακά τόξα χρονικού περιορισμού

Τα δεδομένα στην είσοδο ενός ακολουθιακού κελιού που θα φτάσουν πριν την έλευση του ρολογιού αποθηκεύονται, ενώ αυτά που θα φτάσουν μετά αγνοούνται. Ωστόσο, στην πράξη τα δεδομένα στην είσοδο του κελιού πρέπει να διατηρούνται για ένα χρονικό διάστημα πριν και μετά την αιχμή του ρολογιού για να αποθηκευτούν σωστά. Το χρονικό αυτό διάστημα ονομάζεται περιορισμός (constraint) και το τόξο που σχηματίζεται μεταξύ του σήματος ρολογιού και των δεδομένων, ονομάζεται ακολουθιακό τόξο χρονικού περιορισμού. Τα τόξα ακολουθιακού χρονικού περιορισμού έχουν τέσσερις τύπους περιορισμών. Χρόνος προετοιμασίας, χρόνος συγκράτησης, χρόνος αποκατάστασης και χρόνος απομάκρυνσης, οι έννοιες των οποίων περιεγράφηκαν παραπάνω.

2.4.3.2 Μη ακολουθιακά τόξα χρονικού περιορισμού

Όταν ο σχετιζόμενος ακροδέκτης δεν είναι το ρολόι, τότε το τόξο ονομάζεται μη ακολουθιακό τόξο χρονικού περιορισμού. Κατά τ' άλλα είναι όμοιο με το ακολουθιακό τόξο. Το μη ακολουθιακό τόξο υπάρχει και σε ακολουθιακά και σε συνδυαστικά κελιά. Για παράδειγμα σε ένα flip-flop το μη ακολουθιακό τόξο μπορεί να είναι από το set στο reset, ενώ σε μία πύλη μεταξύ των δύο εισόδων.

2.4.3.3 Τόξα χρονικού περιορισμού στο μοντέλο NLDM

Τώρα θα δούμε πως απεικονίζονται για το NLDM τα τόξα χρονικού περιορισμού στο πρότυπο Liberty. Αυτά τα τόξα βρίσκονται στο γκρουπ "pin" με κατεύθυνση εισόδου (direction: input). Κάθε τέτοιο γκρουπ περιέχει ένα ή περισσότερα γκρουπ με όνομα "timing". Κάθε "timing" γκρουπ περιέχει τις εξής υποομάδες:

- Rise_constraint: τιμές των περιορισμών για την άνοδο του ακροδέκτη εισόδου δεδομένων.
- Falling_constraint: τιμές των περιορισμών για την κάθοδο του ακροδέκτη εισόδου δεδομένων.

Αυτές οι υποομάδες απεικονίζονται σε έναν πίνακα με τιμές για διαφορετικούς συνδυασμούς χρόνων μεταβάσεων της εισόδου δεδομένων (index 1) και χρόνων μεταβάσεων του σχετιζόμενου ακροδέκτη (index 2), όπως δείχνουν οι πίνακες του παρακάτω σχήματος.

	rise_cons	traint"		u	fall_const	traint"	
Index1/Index2	R1	R2	R3	Index1/Index2	R1	R2	R3
D1	CD1R1	CD1R2	CD1R3	D1	CD1R1	CD1R2	CD1R3
D2	CD2R1	CD2R2	CD2R3	D2	CD2R1	CD2R2	CD2R3
D3	CD3R1	CD3R2	CD3R3	D3	CD3R1	CD3R2	CD3R3

Σχήμα	2.9:	Τόξα	χρονικού	περιορισ	μού.
- <u></u>			VL		

Όπου 'D', οι διαφορετικές τιμές των χρόνων μεταβάσεων της εισόδου, όπου 'R' οι διαφορετικές τιμές των χρόνων μεταβάσεων του σχετιζόμενου ακροδέκτη και όπου 'C', ο χρόνος για τον περιορισμό για το ζευγάρι 'D', 'R'.

Αυτό το γκρουπ περιέχει τον σχετιζόμενο ακροδέκτη με την λέξη κλειδί "related_pin", και τον τύπο του περιορισμού με την λέξη κλειδί "timing_type", η οποία μπορεί να πάρει τις εξής τιμές:

- "setup_rising"
- "setup_falling"
- "hold_rising"
- "hold_falling"
- "recovery_rising"
- "recovery falling"
- "removal rising"
- "removal falling"
- "non_seq_setup_rising"
- "non_seq_setup_falling"
- "non_seq_hold_rising"
- "non_seq_hold_falling

2.4.4 Μοντέλο ισχύος

Η αυξανόμενη σημασία για εξοικονόμηση ενέργειας κατέστησε την ανάλυση ισχύος ως ένα σημαντικό βήμα κατά την σχεδίαση. Σε αυτή την παράγραφο θα δούμε το μοντέλο ισχύος NLPM (Non Linear Power Model). Το NLPM μοντέλο είναι μία συλλογή από τις παρακάτω ομάδες:

- Leakage_power: Η στατική κατανάλωση για κάθε συνθήκη ξεχωριστά.
- Cell_leakage_power: Η μέση στατική κατανάλωση του κελιού.
- Internal_power: Η δυναμική κατανάλωση που περιλαμβάνει τις υποομάδες "fall_power" και "rise_power".

2.4.4.1 Στατική ισχύς

Ο ορισμός της στατικής ισχύος (leakage power) εξηγήθηκε παραπάνω και είναι το γινόμενο του ρεύματος που διαρρέει το κελί επί την τάση τροφοδοσίας, όπως δείχνει ο παρακάτω τύπος:

$$P_{\text{static}} = I_{\text{static}} \times V_{\text{DD}}$$

(2.1)

Όπου **V**_{DD} είναι η τάση τροφοδοσίας και **I**_{static} είναι το συνολικό ρεύμα που διαρρέει τι κελί. Το γκρουπ "leakage_power" είναι μέρος του γκρουπ "cell" και υπολογίζεται για κάθε δυνατό συνδυασμό των εισόδων.

2.4.4.2 Στατική ισχύς κελιού

Ορίζεται ως ο μέσος όρος στατικής ισχύος όλων των καταστάσεων όταν το κελί είναι σε κατάσταση αναμονής και υπάρχει στο γκρουπ "cell". Αυτή η ιδιότητα πρέπει να υπάρχει στα κελιά και πρέπει να έχει θετική τιμή. Αν δεν υπάρχει ή είναι αρνητική, τότε χρησιμοποιείται η προκαθορισμένη τιμή που είναι '0'.

2.4.4.3 Δυναμική ισχύς

Η δυναμική ισχύς είναι η ισχύς που καταναλώνει ένα κελί όταν υπάρχει αλλαγή κατάστασης σε κάποια είσοδο ή και έξοδο και βρίσκεται στο γκρουπ "internal_power". Όταν μία διέγερση εφαρμόζεται σε τουλάχιστον μία από τις εισόδους, τότε υπάρχουν δύο περιπτώσεις:

- Να έχουμε αλλαγή στην έξοδο. Αυτή η περίπτωση προκαλεί δύο τύπους κατανάλωσης ισχύος. Την εσωτερική ισχύ (internal power) και την ισχύ μεταγωγής (switching power).
- Να μην έχουμε αλλαγή στην έξοδο. Αυτή η περίπτωση προκαλεί μόνο εσωτερική (internal) κατανάλωση ισχύος. Αυτή καλείται επίσης κρυφή ισχύς, όπως αναφέρθηκε και παραπάνω.

2.4.4.4 Ισχύς μεταγωγής (switching power)

Όπως είπαμε παραπάνω, ισχύς μεταγωγής είναι η ισχύς που καταναλώνεται από το κελί όταν μία αλλαγή σε μία είσοδο προκαλέσει την αλλαγή μίας εξόδου. Όταν η αλλαγή στην είσοδο προκαλεί άνοδο στην έξοδο, η μισή ενέργεια καταναλώνεται για να φορτιστεί η χωρητικότητα φορτίου στο επίπεδο της τάσης τροφοδοσίας (V_{DD}) και η άλλη μισή καταναλώνεται στις αντιστάσεις των μετάλλων στο μονοπάτι από την τροφοδοσία μέχρι την γείωση. Όταν η αλλαγή στην είσοδο προκαλεί κάθοδο στην έξοδο, τότε η χωρητικότητα φορτίου αποφορτίζεται στο επίπεδο της τάσης της γείωσης (V_{SS}) και η ενέργεια που είναι απόθηκευμένη στην χωρητικότητα καταναλώνεται από τις αντιστάσεις των μετάλλων στο μονοπάτι από τις αντιστάσεις των μετάλλων στο μονοπάτι από της τασης την τροφοδοσία μέχρι την γείωση. Όταν η αλλαγή στην είσοδο προκαλεί κάθοδο στην έξοδο, τότε η χωρητικότητα φορτίου αποφορτίζεται στο επίπεδο της τάσης της γείωσης (V_{SS}) και η ενέργεια που είναι αποθηκευμένη στην χωρητικότητα καταναλώνεται από τις αντιστάσεις των μετάλλων στο μονοπάτι προς την γείωση. Ισχύς μεταγωγής είναι το άθροισμα της ισχύος που καταναλώνουν όλες οι έξοδοι ενός κελιού και η μέτρησή της επαναλαμβάνεται για διαφορετικές καταστάσεις της εισόδου, για διαφορετικούς χρόνους μετάβασης των εισόδων και διαφορετικές τέρησης και δεν χρειάζεται να οριστούν ξεχωριστά.

2.4.4.5 Κρυφή ισχύς (Hidden Power)

Όταν εφαρμόζεται μία διέγερση στην είσοδο ενός κελιού, οι εσωτερικές χωρητικότητες του κυκλώματος φορτίζονται και αποφορτίζονται και οι αντιστάσεις μετατρέπουν την ηλεκτρική ενέργεια σε θερμότητα. Το τμήμα της ενέργειας που καταναλώνεται στο εσωτερικό

του κυκλώματος, ονομάζεται κρυφή ισχύς (hidden power). Αυτή η ισχύς καταναλώνεται ασχέτως από το αν υπάρχει αλλαγή στην έξοδο ή όχι. Η κρυφή ισχύς περιλαμβάνει την κατανάλωση ισχύος από το στιγμιαίο βραχυκύκλωμα ανάμεσα στα NMOS και PMOS τρανζίστορς της πύλης και ονομάζεται ισχύς βραχυκυκλώματος.

Για να το καταλάβουμε καλύτερα αυτό, υποθέτουμε έναν αντιστροφέα με PMOS και NMOS τρανζίστορς. Όταν η είσοδός του μεταβάλλεται από '0' σε '1', το PMOS είναι κλειστό και το NMOS ανοιχτό. Ωστόσο για ένα σύντομο χρονικό διάστημα στη μετάβαση των δύο τρανζίστορ και τα δύο τρανζίστορ είναι ανοιχτά. Σε αυτή τη διάρκεια ένα ρεύμα ρέει από την τροφοδοσία προς την γείωση προκαλώντας την ισχύ βραχυκυκλώματος.

Αν η έξοδος μεταβληθεί, η κρυφή ισχύς υπολογίζεται με την αφαίρεση της ισχύος μεταγωγής από τη συνολική δυναμική ισχύ. Η μέτρησή της επαναλαμβάνεται για διαφορετικές καταστάσεις της εισόδου, για διαφορετικούς χρόνους μετάβασης των εισόδων και διαφορετικές χωρητικότητες φορτίου.

Το γκρουπ "internal_power" στο πρότυπο Liberty χρησιμοποιείται για να περιγράψει και την κρυφή ισχύ και την ισχύ μεταγωγής. Η διαφορά μεταξύ τους είναι σε ποιο γκρουπ "pin" θα τοποθετηθούν. Η κρυφή ισχύς τοποθετείται στα "pin" γκρουπ με κατεύθυνση εισόδου και η ισχύς μεταγωγής τοποθετείται στα "pin" γκρουπ με κατεύθυνση εξόδου.

Σε αντίθεση με τα τόξα μεταγωγής ισχύος, τα τόξα κρυφής ισχύος πρέπει να οριστούν ξεχωριστά. Όπως και στα τόξα καθυστέρησης, έτσι κι εδώ το γκρουπ "internal_power" απεικονίζεται σε έναν πίνακα με τιμές για διαφορετικούς συνδυασμούς χρόνων μεταβάσεων της εισόδου δεδομένων (index 1) και χωρητικοτήτων εξόδου (index 2), όπως δείχνουν οι πίνακες του παρακάτω σχήματος.

			"intern	al_power"			
	"rise_pov	wer"			"fall_pov	wer"	
Index1/Index2	C1	C2	C3	Index1/Index2	C1	C2	C3
S1	Es1C1	Es1c2	Es1c3	S1	Es1c1	Es1C2	Es1C3
S2	Es2C1	Es2C2	Es2C3	S2	Es2C1	Es2C2	Es2C3
S3	Essci	Es3c2	Еѕзсз	S 3	Es3C1	Es3C2	Еѕзсз

Σχήμα 2.10: Τόξα δυναμικής ισχύος.

Όπου 'S', οι διαφορετικές μεταβάσεις εισόδου (input slew), όπου 'C' οι διαφορετικές χωρητικότητες εξόδου και όπου 'E' η ενέργεια για το ζευγάρι 'S' και 'C'. Ο παραπάνω πίνακας απεικονίζει την ενέργεια και όχι την ισχύ και έχει διαφορετικές υποομάδες για την ισχύ της ανόδου και την ισχύ της καθόδου.

2.5 Κελιά που θα χαρακτηριστούν

Στην ενότητα αυτή θα δούμε τα τροποποιημένα κελιά που θα χαρακτηριστούν. Να σημειωθεί ότι η τροποποίηση των κελιών δεν είναι αντικείμενο αυτής της εργασίας, παρά μόνο ο χαρακτηρισμός τους, οπότε δε θα γίνει εκτενής αναφορά σε αυτά. Στα συγκεκριμένα κελιά έγινε τροποποίηση στη φυσική σχεδίαση (layout) και προστέθηκαν δακτύλιοι προστασίας (guard rings), ώστε να επιτευχθεί ανθεκτικότητα στο ανοικτοκλείδωμα (latch-up). Τα κελιά τα οποία θα χαρακτηριστούν δεν αποτελούν το σύνολο της βιβλιοθήκης των πρωτότυπων κελιών, αλλά επιλέχθηκαν κάποια βασικά για την υλοποίηση του SPI. Τα κελιά που επιλέχθηκαν για τροποποίηση και χαρακτηρισμό ήταν αυτά που χρησιμοποίησε το εργαλείο σύνθεσης για να παράξει από τον VHDL κώδικα την υλοποίηση του κυκλώματος SPI, με βάση την βιβλιοθήκη κελιών που παρείχε η τεχνολογία του εργοστασίου κατασκευής ημιαγωγών. Η βιβλιοθήκη είναι η **If15adhvt9s** της Lfoundry.

Τα κελιά που θα χαρακτηριστούν είναι τα εξής: AND2_X0d5, NOR2_X0d5, XOR2_X1, INV_X1, CLKBUF_X1, AO21_X0d5, AOI21_X0d5, OAI22_X0d5, HA_X1, DFC_X1, DFNC_X1 και DFCS_X1.

Όπως βλέπουμε στις παρακάτω εικόνες της φυσικής σχεδίασης των κελιών, το πάνω μέρος τους είναι το PMOS κύκλωμα της CMOS διάταξης και το κάτω το NMOS. Αυτά χωρίζονται από τους δακτυλίους προστασίας που σημειώνονται με κίτρινο κύκλο. Με την προσθήκη αυτή χρειάζεται και 2° επίπεδο μετάλλων για να γίνουν οι συνδέσεις μεταξύ των NMOS και PMOS και των εισόδων και εξόδων των κελιών. Το 1° επίπεδο μετάλλων χρησιμοποιείται για την σύνδεση της τροφοδοσίας και της γείωσης και τις εσωτερικές συνδέσεις του PMOS και του NMOS κυκλώματος. Όσο για την επιφάνεια, καλύπτουν την διπλάσια σε σχέση με τα κελιά της lf15adhvt9s, καθώς έχουν το διπλάσιο ύψος λόγω των δακτυλίων προστασίας.

2.5.1 Συνδυαστικά κελιά

Το κελί AND2_X0d5 είναι μία πύλη AND δύο εισόδων με διαστάσεις 2.48 x 9.72 μm και η φυσική σχεδίαση (layout) του απεικονίζεται στο σχήμα 2.11 (A).

Το κελί NOR2_X0d5 είναι μία πύλη NOR δύο εισόδων με διαστάσεις 1.86 x 9.72 μm και η φυσική σχεδίαση του (layout) απεικονίζεται στο σχήμα 2.11 (B).

Το κελί XOR2_X1 είναι μία πύλη XOR δύο εισόδων με διαστάσεις 4.96 x 9.72 μm και η φυσική σχεδίαση του (layout) απεικονίζεται στο σχήμα 2.11 (Γ).

Το κελί INV_X1 είναι ένας αντιστροφέας (πύλη NOT) με διαστάσεις 1.24 x 9.72 μm και η φυσική σχεδίαση του (layout) απεικονίζεται στο σχήμα 2.11 (Δ).

Το κελί CLKBUF_X1 είναι ένας απομονωτής (buffer) και θα χρησιμοποιηθεί στη φυσική σχεδίαση για τη σύνθεση του δέντρου ρολογιού και για τις εξόδους του κυκλώματος και έχει διαστάσεις 1.86 x 9.72 μm. Η φυσική σχεδίαση του (layout) απεικονίζεται στο σχήμα 2.11 (E).



Σχήμα 2.11: Τροποποιημένα layout των κελιών με προσθήκη δακτυλίων προστασίας. Α) AND2_X0d5, B) NOR2_X0d5, Γ) XOR2_X1, Δ) INV_X1 και Δ) CLKBUF_X1.

Το κελί AO21_X0d5 έχει τρείς εισόδους και μία έξοδο. Όπως δείχνει και το σχήμα 2.12, αποτελείται από μία πύλη AND δύο εισόδων οι οποίες είναι οι είσοδοι A1 και B1 του κελιού και μία πύλη OR δύο εισόδων, της οποίας η μία είσοδος είναι η είσοδος B1 του κελιού και η δεύτερη είσοδος πάει στην έξοδο της πύλης AND. Η έξοδος της OR είναι και η έξοδος του κελιού. Το κελί έχει διαστάσεις 3.10 x 9.72 μm και η φυσική σχεδίαση του απεικο-νίζεται στο σχήμα 2.15 (A).



Σχήμα 2.12: Σχηματικό κελιού AO21_X0d5.

Το κελί OAl22_X0d5 έχει τέσσερις εισόδους και μία έξοδο. Όπως δείχνει και το σχήμα 2.13, αποτελείται από μία δύο πύλες OR δύο εισόδων οι οποίες είναι οι είσοδοι A1, A2 και B1, B2 του κελιού και μία πύλη NAND δύο εισόδων, της οποίας η κάθε είσοδος είναι η έξοδος μίας πύλης OR. Η έξοδος της NAND είναι και η έξοδος του κελιού. Το κελί έχει διαστάσεις 3.10 x 9.72 μm και η φυσική σχεδίαση του απεικονίζεται στο σχήμα 2.15 (B).



Σχήμα 2.13: Σχηματικό κελιού OAI22_X0d5.

Το κελί AOI21_X0d5 έχει τρείς εισόδους και μία έξοδο. Όπως δείχνει και το σχήμα 2.14, αποτελείται από μία πύλη AND δύο εισόδων οι οποίες είναι οι είσοδοι A1 και B1 του κελιού και μία πύλη NOR δύο εισόδων, της οποίας η μία είσοδος είναι η είσοδος B1 του κελιού και η δεύτερη είσοδος πάει στην έξοδο της πύλης AND. Η έξοδος της NOR είναι και η έξοδος του κελιού. Το κελί έχει διαστάσεις 2.48 x 9.72 μm και η φυσική σχεδίαση του απεικονίζεται στο σχήμα 2.15 (Γ).



Σχήμα 2.14: Σχηματικό κελιού AOI21_X0d5.



Σχήμα 2.15: Τροποποιημένα layout των κελιών με προσθήκη δακτυλίων προστασίας. Α) ΑΟ21_X0d5, Β) ΟΑΙ22_X0d5 και Γ) ΑΟΙ21_X0d5.

Το κελί HA_X1 είναι ένας ημιαθροιστής με διαστάσεις 6.20 x 9.72 μm και η φυσική σχεδίαση του απεικονίζεται στο σχήμα 2.16.



Σχήμα 2.16: Τροποποιημένο layout του κελιού ΗΑ_Χ1 με προσθήκη δακτυλίων προστασίας.

2.5.2 Ακολουθιακά κελιά

Το κελί DFC_X1 είναι ένα D flip-flop και έχει τρείς εισόδους και δύο εξόδους. Η μία είσοδος είναι το ρολόι χρονισμού ανοδικής αιχμής 'CP', ή δεύτερη είναι η είσοδος δεδομένων 'D' και η τρίτη είναι ασύγχρονο reset 'CDN' που ενεργοποιείται στη λογική κατάσταση '0'. Η μία έξοδος είναι η έξοδος δεδομένων 'Q' και η άλλη είναι αντεστραμμένη έξοδος δεδομένων 'QN'. Το κελί έχει διαστάσεις 13.02 x 9.72 μm και η φυσική σχεδίαση του απεικονίζεται στο σχήμα 2.17.



Σχήμα 2.17: Τροποποιημένο layout του κελιού DFC_X1 με προσθήκη δακτυλίων προστασίας.

Το κελί DFNC_X1 είναι ένα D flip-flop και έχει τρείς εισόδους και δύο εξόδους. Η μία είσοδος είναι το ρολόι χρονισμού καθοδικής αιχμής 'CP', ή δεύτερη είναι η είσοδος δεδομένων 'D' και η τρίτη είναι ασύγχρονο reset 'CDN' που ενεργοποιείται στη λογική κατάσταση '0'. Η μία έξοδος είναι η έξοδος δεδομένων 'Q' και η άλλη είναι αντεστραμμένη έξοδος δεδομένων 'QN'. Το κελί έχει διαστάσεις 11.16 x 9.72 μm και η φυσική σχεδίαση του απεικονίζεται στο σχήμα 2.18.

Το κελί DFCS _X1 είναι ένα D flip-flop και έχει πέντε εισόδους και δύο εξόδους. Η μία είσοδος είναι το ρολόι χρονισμού ανοδικής αιχμής 'CP', ή δεύτερη είναι η είσοδος δεδομένων 'D', η τρίτη είναι η είναι η scan in 'SI', η τέταρτη είσοδος είναι scan enable 'SE' που επιτρέπει να περάσει στην έξοδο την είσοδο 'D' όταν είναι στην λογική κατάσταση '1' ή την είσοδο 'SI' όταν είναι στο '0'. Η πέμπτη είναι ασύγχρονο reset 'CDN' που ενεργοποιείται στη λογική κατάσταση '0'. Η μία έξοδος είναι η έξοδος δεδομένων 'Q' και η άλλη είναι αντεστραμμένη έξοδος δεδομένων 'QN'. Το κελί έχει διαστάσεις 15.5 x 9.72 μm και η φυσική σχεδίαση του απεικονίζεται στο σχήμα 2.19.



Σχήμα 2.18: Τροποποιημένο layout του κελιού DFNC_X1 με προσθήκη δακτυλίων προστασίας.



Σχήμα 2.19: Τροποποιημένο layout του κελιού DFCS_X1 με προσθήκη δακτυλίων προστασίας.

2.6 Χαρακτηρισμός

Σε αυτή την ενότητα θα δούμε λεπτομέρειες για την διαδικασία χαρακτηρισμού και τον προσδιορισμό των χρονικών τόξων. Για τον χαρακτηρισμό χρησιμοποιήσαμε το Liberate της Cadence στην έκδοση 15.1.

Όπως είδαμε και παραπάνω, για να πραγματοποιηθεί ο χαρακτηρισμός χρειάζεται να δώσουμε κάποια αρχικά δεδομένα. Πρώτον τα αρχεία (netlist) που περιέχουν τις εσωτερικές συνδέσεις για το κάθε κελί και τα παρασιτικά του στοιχεία, τα οποία δίνονται από τον σχεδιαστή των κελιών και στην περίπτωση μας είναι σε Spectre μορφή. Μετά είναι το μοντέλο του τρανζίστορ που χρησιμοποιείται για την σχεδίαση των κελιών. Αυτό το αρχείο το παρέχει το εργοστάσιο κατασκευής της συγκεκριμένης τεχνολογίας. Όπως αναφέρθηκε στην προηγούμενη ενότητα, η πρωτότυπη βιβλιοθήκη κελιών είναι η If15adhvt9s της Lfoundry και το τρανζίστορ που χρησιμοποιείται είναι το cmos150_ mos1v8hvt και είναι MOSFET τεχνολογίας 0,15 μm με τάση λειτουργίας στα 1.8V. Τα αρχεία που περιγράφουν το μοντέλο είναι σε μορφή Spectre και είναι το cmos150.scs και το cmos150_mos1v8hvt.scs. Σε αυτά τα αρχεία περιλαμβάνονται οι πληροφορίες για τύπου N και τύπου P τρανζίστορ, για κάθε σημείο διεργασίας.

Τέλος, χρειάζεται το σχεδιότυπο (template) που περιγράφει τα κελιά και τις ρυθμίσεις και το αρχείο με τις εντολές που θα τρέξει το εργαλείο για να πραγματοποιήσει τον χαρακτηρισμό. Τα τόξα χρονισμού και ισχύος τα υπολογίζει αυτόματα το Liberate, παρόλ' αυτά υπάρχει και δυνατότητα να δοθούν και χειροκίνητα από τον χρήστη. Αυτά τα αρχεία είναι γραμμένα σε γλώσσα προγραμματισμού σεναρίων tcl και υπάρχουν στο παράρτημα l.

2.6.1 Δομή του σχεδιότυπου (template)

To template είναι το αρχείο που περιέχει τον προσδιορισμό των κελιών, τις παραμέτρους του χαρακτηρισμού και διάφορες ρυθμίσεις για το Liberate. Οι παράμετροι που χρησιμοποιήσαμε είναι οι εξής:

- Τα όρια που μετριέται ο χρόνος μετάβασης των σημάτων εισόδου και εξόδου ορίζονται από το 10% έως το 90% για την άνοδο και από το 90% έως το 10% για την κάθοδο.
- Η καθυστέρηση (delay) σε ένα ζευγάρι ακροδεκτών ορίζεται να μετριέται από το 50% του σήματος εισόδου έως το 50% του σήματος εξόδου.
- Ο ελάχιστος χρόνος μετάβασης εισόδου (min input slew) ορίζεται σε 0,01 ns.
- Ο μέγιστος χρόνος μετάβασης εισόδου (max input slew) ορίζεται σε 2,5 ns.
- Η ελάχιστη χωρητικότητα εξόδου του κελιού ορίζεται σε 1 fF.
- Η μέγιστη χωρητικότητα εξόδου του κελιού ορίζεται από το Liberate.

Επόμενο βήμα είναι να ορίσουμε το μέγεθος των πινάκων, δηλαδή το πλήθος των μετρήσεων για διαφορετικά ζευγάρια χρόνου μετάβασης εισόδου και χωρητικότητας εξόδου, όπως αναφέραμε σε προηγούμενη ενότητα. Όσο μεγαλύτερος ο πίνακας, τόσο ακριβέστερα θα είναι τα αποτελέσματα, αλλά σαν συνέπεια θα έχει το μεγαλύτερο μέγεθος του αρχείου και μεγαλύτερο χρόνο της διαδικασίας του χαρακτηρισμού.

τον χρόνων μετάβασης του ακροδέκτη με τον χρονικό περιορισμό (set, reset) και το index2 δηλώνει τις τιμές τον χρόνων μετάβασης του σχετιζόμενου ακροδέκτη. Το Liberate με βάση τις ελάχιστες και μέγιστες τιμές που ορίσαμε για τους χρόνους μετάβασης και τις χωρητικότητες εξόδου, γεμίζει τα index1 και index2, με τιμές εντός των ορίων που δηλώσαμε.

Τέλος προσδιορίζουμε τα κελιά. Αυτό γίνεται δηλώνοντας το όνομα του κελιού και τους εισόδους και εξόδους που έχει. Όλα τα υπόλοιπα τα κάνει αυτόματα το Liberate.

Για τις ανάγκες της εργασίας κάναμε χαρακτηρισμό για διαφορετικές συνθήκες λειτουργίας (ακρότατα σημεία και θερμοκρασίες). Συγκεκριμένα κάναμε για τα σημεία TT, FF, SS, SF και FS και για θερμοκρασίες -40, 25 και 70 για το κάθε σημείο, δηλαδή συνολικά παράχθηκαν δεκαπέντε αρχεία.

2.7 Αποτελέσματα χαρακτηρισμού

Για να πάρουμε μία εικόνα των κελιών που φτιάξαμε, θα συγκρίνουμε τη βιβλιοθήκη που φτιάξαμε με την πρωτότυπη της Lfoundry, για ακρότατο σημείο TT σε 25 C° και επίσης μία σύγκριση μεταξύ ακρότατου σημείο FF για -40 C° και ακρότατου σημείο SS για 70 C°, που αποτελούν τις ακραίες συνθήκες της βιβλιοθήκης που φτιάξαμε. Στα διαγράμματα που ακολουθούν βλέπουμε τις τιμές των χαρακτηριστικών (καθυστέρηση διάδοσης, χρόνο μετάβασης εξόδου, κατανάλωση ενέργειας και χρονικούς περιορισμούς για τα flipflop) για το κάθε κελί. Στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκη της lfoundry και στον κάθετο οι τιμές για την τροποποιημένη βιβλιοθήκη. Κάθε σημείο στο διάγραμμα αντιστοιχεί στην τιμή για το ίδιο ζευγάρι τιμών (χρόνος μετάβασης εισόδου χωρητικότητα εξόδου) των δύο συγκρινόμενων βιβλιοθηκών και σε σχέση με τη διαγώνια γραμμή, δείχνει σε ποια βιβλιοθήκη αντιστοιχεί η μεγαλύτερη τιμή για το κάθε ζευγάρι.

2.7.1 Αποτελέσματα για το κελί AND2_X0d5

Στο σχήμα 2.20 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 17.39 pW έναντι 11,21 pW της πρωτότυπης και είναι αυξημένη κατά 6.17 pW, δηλαδή 55,5%.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff% Ou	tliers
	++-	***************	+	+	+	*************	
leakage(pW)	6	6.178236	55.51%	9.23%	9.084707	65.48%	6

Σχήμα 2.20: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d5.

Στο σχήμα 2.21 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 16,45 pW για σημείο FF, έναντι 18 pW για SS. Υπάρχει μια διαφορά 1,6 pW που είναι της τάξης του 9,8%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
+	+-	++	++		+	************	***********	*********
1	leakage(pW)	6 I	1.595817	9.81%	1.30%	2.116800	11.56%	6
£	reakage (pa) 1		1.000011		1.308 1	2.1100000 1		
Mark	er leakage ourli	er. May the.	2 116900	Poul # · ·	+ May Dalr	11 56% 000		

Σχήμα 2.21: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί AND2_X0d5.

Στο σχήμα 2.22 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο ΤΤ. Το τροποποιημένο κελί βλέπουμε ότι έχει μικρότερους χρόνος καθυστέρησης. Η μέση διαφορά μεταξύ τους είναι στα 0,017 ns που

αντιστοιχεί σε ποσοστό 6,25%, με την μέγιστη διαφορά να φτάνει τα 0,040 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,08 ns έως 1,1 ns.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,054 ns έως 0,806 ns, ενώ στο σημείο SS από 0,113 ns έως 1,61 ns. Όπως δείχνει και το σχήμα 2.23, η μέση διαφορά μεταξύ τους είναι 0,261 ns που αντιστοιχεί σε 54% και η μέγιστη διαφορά είναι 1,07 ns.



Σχήμα 2.22: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d5.

Data Type	Entries	Avg Diff	Avg Diffs	Sigma%	Max Diff	Max Diff%	Outliers
delay(ns)	400 1	-0.261439	-54.31%	8.56%	-1.071463	-91.80%	400

Σχήμα 2.23: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί AND2_X0d5.

Στο σχήμα 2.24 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά βλέπουμε ότι έχουν ελάχιστη διαφορά. Το τροποποιημένο κελί είναι λίγο πιο γρήγορο με την μέση διαφορά μεταξύ τους να είναι στα 0,002 ns, που αντιστοιχεί σε ποσοστό 2,4%, με την μέγιστη διαφορά να φτάνει τα 0,010 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,05 ns έως 1,9 ns για την άνοδο και από 0,04 ns έως 0,3 ns για την κάθοδο.



Σχήμα 2.24: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,035 ns έως 1,02 ns για την άνοδο και από 0,028 ns έως 0,055 ns για την κάθοδο. Ενώ στο σημείο SS από 0,075 ns έως 2,5 ns για την άνοδο και από 0,060 ns έως 1,17 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.25, η μέση διαφορά μεταξύ τους είναι 0,256 ps που αντιστοιχεί σε 51% και η μέγιστη διαφορά είναι 1,49 ns.

1	Data Type	Entries	Avg Diff	1	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
	trans(ns)	400	-0.256803	1	-51.35%	6.13%	-1.494193	-59.55%	400

Σχήμα 2.25: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί AND2_X0d5.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο. Το τροποποιημένο κελί βλέπουμε ότι έχει ελαφρώς μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να κυμαίνεται στα 0,000215 pJ που αντιστοιχεί σε ποσοστό 79,32%. Ενδεικτικά οι τιμές κυμαίνονται από 0,008 pJ έως 0,016 pJ.



Σχήμα 2.26: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AND2_X0d5.

E.	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Difft	Outliers
t		t	t	++		t		
1	power(pJ)	440	0.002761	29.05%	38.81%	0.014101	171,94%	237

Σχήμα 2.27: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί AND2_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,001 pJ έως 0,026 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,013 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την

SS. Όπως δείχνει και το σχήμα 2.27, η μέση διαφορά μεταξύ τους είναι 0,0027 pJ που αντιστοιχεί σε 29%.

2.7.2 Αποτελέσματα για το κελί NOR2_X0d5

Στο σχήμα 2.28 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 17.39 pW έναντι 10,17 pW της πρωτότυπης και είναι αυξημένη κατά 6.95 pW, δηλαδή 50,6%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
	leakage(pW)	6 1	3.217459 (50.59%	17.22%	6.096207 1	80,83%	6

Σχήμα 2.28: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5.

Στο σχήμα 2.29 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 9,55 pW για σημείο FF, έναντι 10,5 pW για SS. Υπάρχει μια διαφορά 1 pW που είναι της τάξης του 8,33%, δηλαδή σε σημείο FF έχουμε χαμηλότερη κατανάλωση.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
leakage(pW)	6 1	-1.009310	-8.33%	3.52%	-2.599500	-13.12%	6

Σχήμα 2.29: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί NOR2_X0d5.

Στο σχήμα 2.30 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Τα δύο κελιά έχουν ελάχιστη διαφορά με την μέση διαφορά μεταξύ τους να είναι στα 0,002 ns που αντιστοιχεί σε ποσοστό 1,26%, με ταχύτερο το τροποποιημένο κελί και η μέγιστη διαφορά είναι 0,012 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,05 ns έως 2,6 ns.



Σχήμα 2.30: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,013 ns έως 1,07 ns, ενώ στο σημείο SS από 0,022 ns έως 1,90 ns. Όπως δείχνει και το σχήμα 2.31, η μέση διαφορά μεταξύ τους είναι 0,180 ns που αντιστοιχεί σε 51% και η μέγιστη διαφορά είναι 0,88 ns.

Hole Hoge deral containt ignocor closting, rereat to	NOR2	X0d5	delay	SUMMARY	(abstol	0.001ns,	reltol	18)
--	------	------	-------	---------	---------	----------	--------	-----

1	Data Type	Entries	Avg Diff	Avg Diff&	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	400	-0.180847	-51.18% (15.23%	-0.886151	-158.91%	400
* Worst	delay outlier:	Max Abs:	-0.886151, Ro	w # : 300;	Max Rel:	-158.91%, Row #	: 191	++

Σχήμα 2.31: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί NOR2_X0d5.

Στο σχήμα 2.32 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά έχουν ελάχιστη διαφορά με την μέση διαφορά μεταξύ τους να είναι στα 0,004 ns που αντιστοιχεί σε ποσοστό 1,23%, με ταχύτερο το τροποποιημένο κελί και η μέγιστη διαφορά να φτάνει τα 0,031 ns που αντιστοιχεί σε 4,6%. Ενδεικτικά οι τιμές κυμαίνονται από 0,100 ns έως 3,9 ns για την άνοδο και από 0,05 ns έως 1,3 ns για την κάθοδο.



Σχήμα 2.32: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,065 ns έως 1,23 ns για την άνοδο και από 0,015 ns έως 0,724 ns για την κάθοδο. Ενώ στο σημείο SS από 0,150 ns έως 2,42 ns για την άνοδο και από 0,024 ns έως 0,728 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.33, η μέση διαφορά μεταξύ τους είναι 0,245 ns που αντιστοιχεί σε 41% και η μέγιστη διαφορά είναι 1,40 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	400 1	-0.245798	-41,49% (15.86%	-1.405210 1	-61.15%	400



Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως δείχνει και το σχήμα 2.34. Το τροποποιημένο κελί έχει ελαφρώς μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να κυμαίνεται στα 0,0005 pJ που αντιστοιχεί σε ποσοστό 36%. Ενδεικτικά οι τιμές κυμαίνονται από 0,008 pJ έως 0,016 pJ.



Σχήμα 2.34: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί NOR2_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,002 pJ έως 0,031 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,009 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.35, η μέση διαφορά μεταξύ τους είναι 0,003 pJ που αντιστοιχεί σε 60%.

NOR2	_X0d5 power SUN	(MARY (absto	1 0.001pJ, relta	ol 1%)				
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
	power(pJ)	440	0.003089	60.27%	117.15%	0.021137	591.25%	266

Σχήμα 2.35: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί NOR2_X0d5.

2.7.3 Αποτελέσματα για το κελί XOR2_X1

Στο σχήμα 2.36 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 35 pW έναντι 27,3 pW της πρωτότυπης και είναι αυξημένη κατά 7.7 pW, δηλαδή 32,6%.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma	Max Diff	Max Diff%	Outliers
leakage(pW)	6 1	7.719182	32.65% (16.33%	7.906300	69.85%	6

Σχήμα 2.36: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X1.

Στο σχήμα 2.37 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 30,1 pW για σημείο FF, έναντι 38,9 pW για SS. Υπάρχει μια διαφορά 8,8 pW που είναι της τάξης του 22,1%, δηλαδή σε σημείο FF έχουμε χαμηλότερη κατανάλωση.

Data Type	Entries	Avg Diff	Avg Diff% (Sigma%	Max Diff	Max Diff%	Outliers
leakage(pW)	6 1	-8.824717	-22.13% (3.90% 1	-12.229800	-27.51%	6

Σχήμα 2.37: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί XOR2_X1.

Στο σχήμα 2.38 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Τα δύο κελιά έχουν μικρή διαφορά με την μέση διαφορά μεταξύ τους να είναι 0,006 ns που αντιστοιχεί σε ποσοστό 1,6%,με ταχύτερο το πρωτότυπο κελί και η μέγιστη διαφορά να φτάνει τα 0,027 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,03 ns έως 2,6 ns.



Σχήμα 2.38: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 3 ps έως 1,1 ns, ενώ στο σημείο SS από 0,100 ns έως 1,9 ns. Όπως δείχνει και το σχήμα 2.39, η μέση διαφορά μεταξύ τους είναι 0,207 ns που αντιστοιχεί σε 50% και η μέγιστη διαφορά είναι 0,93 ns.

	Data Type	Entries	Avg Diff	Avg Difft	Sigma%	Max Diff	Max Difft	Outliers
1	delay(ns)	800	-0.207767	-50.64%	9.40%	-0.935344	-98,61%	800

Σχήμα 2.39: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί XOR2_X1.

Στο σχήμα 2.40 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Το τροποποιημένο κελί είναι λίγο πιο αργό με την μέση διαφορά μεταξύ τους να κυμαίνεται στα 0,008 ns που αντιστοιχεί σε ποσοστό 3,15%, με την μέγιστη διαφορά να φτάνει τα 0,044 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,11 ns έως 3,9 ns για την άνοδο και από 0,049 ns έως 1,87 ns για την κάθοδο.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,085 ns έως 1,28 ns για την άνοδο και

από 0,036 ns έως 0,88 ns για την κάθοδο. Ενώ στο σημείο SS από 0,200 ns έως 2,5 ns για την άνοδο και από 0,065 ns έως 1,22 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.33, η μέση διαφορά μεταξύ τους είναι 257 ps που αντιστοιχεί σε 47% και η μέγιστη διαφορά είναι 1,37 ns.



Σχήμα 2.40: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X1.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	800 1	-0.257688	-47,47% 1	10.76%	-1.372250	-59.82%	800

Σχήμα 2.41: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί XOR2_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως δείχνει το σχήμα 2.42. Το τροποποιημένο κελί έχει μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,002 pJ που αντιστοιχεί σε ποσοστό 30%. Ενδεικτικά οι τιμές κυμαίνονται από 0,008 pJ έως 0,071 pJ.



Σχήμα 2.42: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί XOR2_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,002 pJ έως 0,096 pJ, ενώ στο σημείο SS από 0,004 pJ έως 0,06 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.43, η μέση διαφορά μεταξύ τους είναι 0,007 pJ που αντιστοιχεί σε 27,5%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma∜	Max Diff	Max Diff%	Outliers
1	power(pJ)	800 1	0.007619 (27.50% 1	56.714	0.038482	394.81%	778

Σχήμα 2.43: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί XOR2_X1.

2.7.4 Αποτελέσματα για το κελί INV_X1

Στο σχήμα 2.44 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 10,6 pW έναντι 7,6 pW της πρωτότυπης και είναι αυξημένη κατά 3 pW, δηλαδή 46,3%.

Data 1	Type	Entries	Avg Diff	Avg Diff%	Sigma≬	Max Diff	Max Diff%	Outliers
leakage	((Rq)	4	3.052988	46.35%	19.34%	3.103330	78.47%	4

Σχήμα 2.44: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X1.

Στο σχήμα 2.45 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 9 pW για σημείο FF, έναντι 12 pW για SS. Υπάρχει μια διαφορά 3 pW που είναι της τάξης του 24,2%, δηλαδή σε σημείο FF έχουμε χαμηλότερη κατανάλωση.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	leakage(pW)	4	-2.973243	-24.20%	2.174	-4.273000	-26.70%	4

Σχήμα 2.45: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί INV_X1.

Στο σχήμα 2.47 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Τα δύο κελιά έχουν σχεδόν μηδενική διαφορά με την μέση διαφορά μεταξύ τους να είναι 0,001 ns που αντιστοιχεί σε ποσοστό 0,45%, με ταχύτερο το πρωτότυπο κελί και η μέγιστη διαφορά είναι 0,019 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,024 ns έως 1,65 ns.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,011 ns έως 1,16 ns, ενώ στο σημείο SS από 0,018 ns έως 1,94 ns. Όπως δείχνει και το σχήμα 2.46, η μέση διαφορά μεταξύ τους είναι 0,146 ns που αντιστοιχεί σε 47,8% και η μέγιστη διαφορά είναι 0,786 ns.

	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	200	-0.146716	-47.80%	14.48%	-0.785979	-139.94%	200

Σχήμα 2.46: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί INV_X1.



Σχήμα 2.47: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X1.

Στο σχήμα 2.48 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Η μέση διαφορά των δύο κελιών είναι σχεδόν μηδενική, της τάξης 0,04% με την μέγιστη διαφορά να φτάνει τα 0,026 ns που αντιστοιχεί σε 6,84%, με ελαφρώς ταχύτερο το πρωτότυπο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,03 ns έως 2,1 ns για την άνοδο και από 0,02 ns έως 1,32 ns για την κάθοδο.



Σχήμα 2.48: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,025 ns έως 1,34 ns για την άνοδο και από 0,01 ns έως 1,03 ns για την κάθοδο. Ενώ στο σημείο SS από 0,046 ns έως 2,48 ns για την άνο-δο και από 0,019 ns έως 1,35 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.49, η μέση διαφορά μεταξύ τους είναι 0,2 ns που αντιστοιχεί σε 39% και η μέγιστη διαφορά είναι 1,33 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	200 1	-0.206684	-39.52%	1 13.11%	-1.339306	-59.23%	200

Σχήμα 2.49: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί INV_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως δείχνει το σχήμα 2.50. Το τροποποιημένο κελί έχει μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να κυμαίνεται στα 0,003 pJ που αντιστοιχεί σε ποσοστό 65%. Ενδεικτικά οι τιμές κυμαίνονται από 0,001 pJ έως 0,033 pJ.



Σχήμα 2.50: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί INV_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,0008 pJ έως 0,063 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,018 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.51, η μέση διαφορά μεταξύ τους είναι 0,007 pJ που αντιστοιχεί σε 91,1%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	200 1	0.006890 1	91.10%	168.50%	0.045059	650,15%	162
Wors	t power outlies	: Max Abs;	0.045059, Ro	w≢: 91;	Max Rel:	650.15%, Row #	: 159	+

Σχήμα 2.51: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί INV_X1.

2.7.5 Αποτελέσματα για το κελί CLKBUF_X1

Στο σχήμα 2.52 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 18,17 pW έναντι 12,09 pW της πρωτότυπης και είναι αυξημένη κατά 6,08 pW, δηλαδή 50,2%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma% (Max Diff	Max Diff%	Outliers
1	leakage(pW)	4	6.081116	50.27%	0.15%	6.097992 1	50.54%	4

Σχήμα 2.52: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X1.

Στο σχήμα 2.53 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 16,03 pW για σημείο FF, έναντι 20,14 pW για SS. Υπάρχει μια διαφορά 4,11 pW που είναι της τάξης του 20,4%, δηλαδή σε σημείο FF έχουμε χαμηλότερη κατανάλωση.

Data Typ	e I	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
leakage (p)	1) (4 1	-4.114425	-20.43%	0.01%	-4.114500	-20.44%	4

Σχήμα 2.53: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί CLKBUF_X1.

Στο σχήμα 2.54 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο ΤΤ. Το τροποποιημένο κελί είναι ελάχιστα ταχύτερο με την μέση διαφορά μεταξύ τους να είναι 0,004 ns που αντιστοιχεί σε ποσοστό 2,45% και η μέγιστη διαφορά είναι 0,01 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,06 ns έως 1,2 ns.



Σχήμα 2.54: Διάγραμμα σύγκρισης καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,030 ns έως 0,76 ns, ενώ στο σημείο SS από 0,075 ns έως 1,53 ns. Όπως δείχνει και το σχήμα 2.55, η μέση διαφορά μεταξύ τους είναι 0,201 ns που αντιστοιχεί σε 49,7% και η μέγιστη διαφορά είναι 0,897 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	200	-0.201042	-49,71%	5.44%	-0.897231	-65.07%	200

Σχήμα 2.55: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί CLKBUF_X1.

Στο σχήμα 2.56 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT και όπως διακρίνεται η διαφορά μεταξύ των δύο κελιών είναι απειροελάχιστη. Το τροποποιημένο κελί είναι ελάχιστα πιο γρήγορο με την μέση διαφορά μεταξύ τους να κυμαίνεται στα 0,001 ns που αντιστοιχεί σε ποσοστό 0,23%, με την μέγιστη διαφορά να φτάνει τα 0,015 ns. Ενδεικτικά οι τιμές κυμαίνονται από 0,03 ns έως 1,8 ns για την άνοδο και από 0,025 ns έως 1,45 ns για την κάθοδο.



Σχήμα 2.56: Διάγραμμα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,022 ns έως 1,04 ns για την άνοδο και από 0,019 ns έως 0,945 ns για την κάθοδο. Ενώ στο σημείο SS από 0,048 ns έως 2,5 ns για την άνοδο και από 0,038 ns έως 1,73 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.57, η μέση διαφορά μεταξύ τους είναι 0,257 ns που αντιστοιχεί σε 49,3% και η μέγιστη διαφορά είναι 1,45 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma% (Max Diff	Max Diff%	Outliers
+		+						
1	trans(ns)	500 1	-0.257274 1	-49.30%	0-43%	-1+400303	-28-504	200

Σχήμα 2.57: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί CLKBUF_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως δείχνει το σχήμα 2.59. Το τροποποιημένο κελί έχει μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να κυμαίνεται στα 0,001 pJ που αντιστοιχεί σε ποσοστό 27,7%. Ενδεικτικά οι τιμές κυμαίνονται από 0,013 pJ έως 0,040 pJ.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,013 pJ έως 0,063 pJ, ενώ στο σημείο SS από 0,012 pJ έως 0,028 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.58, η μέση διαφορά μεταξύ τους είναι 0,007 pJ που αντιστοιχεί σε 39,7%.

1	Data Type	Entries	Avg Diff	Avg Diff% (Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	200 1	0.007646	39.69%	57.46%	0.038174	224.81%	150
Worst	power outlier	: Max Abs:	0,038174, R	ow ♦ : 141; M	ax Rel:	224.81%, Row # :	41	

Σχήμα 2.58: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί CLKBUF_X1.



Σχήμα 2.59: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί CLKBUF_X1.

2.7.6 Αποτελέσματα για το κελί ΗΑ_Χ1

Στο σχήμα 2.60 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 51 pW έναντι 36,60 pW της πρωτότυπης και είναι αυξημένη κατά 14,4 pW, δηλαδή 40,7%.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
	+	+		· · · · · · · · · · · · · · · · · · ·	*-	************	
leakage(pW)	6 1	14,406996	40,778 1	11.31%	17.594609	65.278	6

Σχήμα 2.60: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ1.

Στο σχήμα 2.61 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 46 pW για σημείο FF, έναντι 54,7 pW για SS. Υπάρχει μια διαφορά 8,7 pW που είναι της τάξης του 15,8%, δηλαδή σε σημείο FF έχουμε χαμηλότερη κατανάλωση.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	leakage(pW)	6	-8.706250	-15,82%	1.95%	-10.596500	-18.23% (6

Σχήμα 2.61: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί HA_X1.

Στο σχήμα 2.62 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Σε κάποιες περιπτώσεις είναι ταχύτερο το ένα κελί και σε κάποιες το άλλο. Η μέση διαφορά μεταξύ τους είναι 0,011 ns που αντιστοιχεί σε ποσοστό 4,04%, με ταχύτερο να είναι το τροποποιημένο. Ενδεικτικά οι τιμές κυμαίνονται από 0,054 ns έως 2,85 ns.



Σχήμα 2.62: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,01 ns έως 1,11 ns, ενώ στο σημείο SS από 0,031 ns έως 1,91 ns. Όπως δείχνει και το σχήμα 2.63, η μέση διαφορά μεταξύ τους είναι 0,216 ns που αντιστοιχεί σε 51,7% και η μέγιστη διαφορά είναι 0,99 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	1200	-0.216499	-51.77% (11.16%	-0,995794	-133,19%	1200

Σχήμα 2.63: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί HA_X1.

Στο σχήμα 2.64 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,0003 ns που αντιστοιχεί σε ποσοστό 1,49% με ταχύτερο το τροποποιημένο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,03 ns έως 4, ns για την άνοδο και από 0,04 ns έως 1,9 ns για την κάθοδο.



Σχήμα 2.64: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί HA_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,03 ns έως 1,28 ns για την άνοδο και από 0,02 ns έως 0,8 ns για την κάθοδο. Ενώ στο σημείο SS από 0,06 ns έως 1,5 ns για την άνοδο και από 0,04 ns έως 1,1 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.65, η μέση διαφορά μεταξύ τους είναι 0,242 ns που αντιστοιχεί σε 47,8% και η μέγιστη διαφορά είναι 1,47 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	1200	-0.242263	-47.86%	10.10%	-1.478671	-59.95%	1200

Σχήμα 2.65: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί HA_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο ΤΤ, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.66. Το τροποποιημένο κελί έχει ελάχιστα μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,0001 pJ που αντιστοιχεί σε 184%. Ενδεικτικά οι τιμές κυμαίνονται από 0,002 pJ έως 0,045 pJ.



/home/lambrop/LF15Ai/PDK_LF15Ai_V1_2_0/digital/lf15adhvt9s/liberty/lf15adhvt9s_typical.lib

Σχήμα 2.66: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΗΑ_Χ1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,006 pJ έως 0,051 pJ, ενώ στο σημείο SS από 0,002 pJ έως 0,039 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.67, η μέση διαφορά μεταξύ τους είναι 0,005 pJ που αντιστοιχεί σε 42,6%.

1	Data Type	Entries (Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	1200	0.005755	42.60%	77.94%	0.046364	1403.95%	947

Σχήμα 2.67: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί HA_X1.

2.7.7 Αποτελέσματα για το κελί AO21_X0d5

Στο σχήμα 2.68 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 23,64 pW έναντι 16,70 pW της πρωτότυπης και είναι αυξημένη κατά 6,94 pW, δηλαδή 44,1%.

+		t		++-		······		·
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
÷		+				·····		
1	leakage(pW)	10	6,941233	44.11%	16.00%	9,060344 1	66.86%	10
Wor	ot leakage out!	ar: May Bhe-	0.06034A	Poul # -	A: Nav Dol:	EE RES DAU	+ - 4	

Σχήμα 2.68: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΑΟ21_Χ0d5.

Στο σχήμα 2.69 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 22.37 pW για σημείο FF, έναντι 24,30 pW για SS, δηλαδή στο σημείο FF έχουμε χαμηλότερη κατανάλωση. Υπάρχει μια διαφορά 1,9 pW που είναι της τάξης του 8%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma≹	Max Diff	Max Diff%	Outliers
1	leakage(pW)	10 1	-1,936550	-7,99%	0,99%	-2.191300	-9.22%	10

Σχήμα 2.69: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί AO21_X0d5.

Στο σχήμα 2.70 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Το τροποποιημένο κελί είναι ταχύτερο με την μέση διαφορά μεταξύ τους να είναι 0,034 ns που αντιστοιχεί σε ποσοστό 9,7%. Ενδεικτικά οι τιμές κυμαίνονται από 0,07 ns έως 1,25 ns.



Σχήμα 2.70: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AO21_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,02 ns έως 0,93 ns, ενώ στο σημείο SS από 0,09 ns έως 1,64 ns. Όπως δείχνει και το σχήμα 2.71, η μέση διαφορά μεταξύ τους είναι 0,308 ns που αντιστοιχεί σε 55,6% και η μέγιστη διαφορά είναι 1,09 ns.
Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
 delay(ns)	1000	-0.308256	-55.65%	8.23%	-1.094813	-95.34%	1000

Σχήμα 2.71: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί AO21_X0d5.

Στο σχήμα 2.72 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,007 ns που αντιστοιχεί σε ποσοστό 5,7% με ταχύτερο το τροποποιημένο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,05 ns έως 1,9 ns για την άνοδο και από 0,06 ns έως 0,9 ns για την κάθοδο.



Σχήμα 2.72: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AO21_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,035 ns έως 1,01 ns για την άνοδο και από 0,035 ns έως 0,56 ns για την κάθοδο. Ενώ στο σημείο SS από 0,07 ns έως 2,5 ns για την άνοδο και από 0,078 ns έως 1,04 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.73, η μέση διαφορά μεταξύ τους είναι 0,266 ns που αντιστοιχεί σε 52,6% και η μέγιστη διαφορά είναι 1,48 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	1000	-0.266026	-52.63%	5.31%	-1.485721	-59.57%	1000

Σχήμα 2.73: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί ΑΟ21_Χ0d5.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.74. Το τροποποιημένο κελί έχει ελάχιστα μικρότερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,0006 pJ που αντιστοιχεί σε 49,3%. Ενδεικτικά οι τιμές κυμαίνονται από 0,005 pJ έως 0,018 pJ.



Σχήμα 2.74: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΑΟ21_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,0007 pJ έως 0,027 pJ, ενώ στο σημείο SS από 0,0007 pJ έως 0,014 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.75, η μέση διαφορά μεταξύ τους είναι 0,002 pJ που αντιστοιχεί σε 24%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	1140 (0.002647	24.00%	30.74% [0.013906	145.36%	710
Wors	t power outlier	: Max Abs:	0.013906, Ro	w # : 317; M	ax Rel:	145.36%, Row # :	36	+

Σχήμα 2.75: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί AO21_X0d5.

2.7.8 Αποτελέσματα για το κελί AOI21_X0d5

Στο σχήμα 2.76 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 16,12 pW έναντι 12,31 pW της πρωτότυπης και είναι αυξημένη κατά 3,8 pW, δηλαδή 34,6%.

E	ata Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
i lea	kage (pW)	10	3.811074	34.67%	20.88%	5.925491	83.41%	10

Σχήμα 2.76: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί ΑΟΙ21_Χ0d5.

Στο σχήμα 2.77 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 15.09 pW για σημείο FF, έναντι 16,74 pW για SS, δηλαδή στο σημείο FF έχουμε χαμηλότερη κατανάλωση. Υπάρχει μια διαφορά 1,64 pW που είναι της τάξης του 9%.

Data Type	Entries (Avg Diff	Avg Diff%	Sigma%	Max Diff (Max Diff%	Outliers
leakage(pW)	10	-1.644520	-9.03%	4.40%	-3.165100	-15.36%	10

Σχήμα 2.77: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί AOI21_X0d5.

Στο σχήμα 2.78 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο ΤΤ. Τα δύο κελιά έχουν απειροελάχιστη διαφορά, με την μέση διαφορά μεταξύ τους να είναι 0,002 ns που αντιστοιχεί σε ποσοστό 1,18%. Ενδεικτικά οι τιμές κυμαίνονται από 0,02 ns έως 2,65 ns.



Σχήμα 2.78: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AOI21_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,02 ns έως 1,08 ns, ενώ στο σημείο SS από 0,03 ns έως 1,91 ns. Όπως δείχνει και το σχήμα 2.79, η μέση διαφορά μεταξύ τους είναι 0,19 ns που αντιστοιχεί σε 50,6% και η μέγιστη διαφορά είναι 0,9 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	1000	-0.193853	-50.58%	1 11.17%	-0.912133	-114.56%	1000

Σχήμα 2.79: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί AOI21_X0d5.

Στο σχήμα 2.80 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,003 ns που αντιστοιχεί σε ποσοστό 1,2% με ταχύτερο το τροποποιημένο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,1 ns έως 3,98 ns για την άνοδο και από 0,03 ns έως 1,85 ns για την κάθοδο.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,07 ns έως 1,25 ns για την άνοδο και από 0,025 ns έως 0,85 ns για την κάθοδο. Ενώ στο σημείο SS από 0,16 ns έως 2,5 ns για την άνοδο και από 0,042ns έως 1,17 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.81, η μέση διαφορά μεταξύ τους είναι 0,252 ns που αντιστοιχεί σε 42,5% και η μέγιστη διαφορά είναι 1,4 ns.



Σχήμα 2.80: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AOI21_X0d5.

1	Data Type	Entries	Avg Diff	Av	g Diff%	1	Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	1000	-0.252746		-42.58%	1	14.94%	-1.399703	-61.16%	1000

Σχήμα 2.81: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί ΑΟΙ21_Χ0d5.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.82. Το τροποποιημένο κελί έχει ελάχιστα μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,0003 pJ που αντιστοιχεί σε 12,3%. Ενδεικτικά οι τιμές κυμαίνονται από 0,001 pJ έως 0,019 pJ.



Σχήμα 2.82: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί AOI21_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,001 pJ έως 0,033 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,013 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.83, η μέση διαφορά μεταξύ τους είναι 0,003 pJ που αντιστοιχεί σε 36,4%.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Dutliers
power(pJ)	1 1140	0.002933	1 36.37%	72.86%	1 0.020158 1	462.42%	645

Σχήμα 2.83: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί AOI21_X0d5.

2.7.9 Αποτελέσματα για το κελί OAl22_X0d5

Στο σχήμα 2.84 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 17,40 pW έναντι 12,77 pW της πρωτότυπης και είναι αυξημένη κατά 4,62 pW, δηλαδή 41,9%.

1	Data Type	Entries	Avg Diff (Avg Diff%	Sigma%	Max Diff	Max Diff*	Outliers
1	leakage(pW)	18	4.627633	41.87%	18.07%	5.806407	81.48%	18 1

Σχήμα 2.84: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί OAI22_X0d5.

Στο σχήμα 2.85 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 16.35 pW για σημείο FF, έναντι 18,03 pW για SS, δηλαδή στο σημείο FF έχουμε χαμηλότερη κατανάλωση. Υπάρχει μια διαφορά 1,68 pW που είναι της τάξης του 8,8%.

Į	Data Type	Entries	Avg Diff	Avg Diff%	Sigma≬	Max Diff	Max Diff%	Outliers
1	leakage(pW)	18	-1.600439	-8.80%	3.23%	-2.890000	-13.33%	18

Σχήμα 2.85: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί OAl22_X0d5.

Στο σχήμα 2.87 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο ΤΤ. Τα δύο κελιά έχουν απειροελάχιστη διαφορά, με την μέση διαφορά μεταξύ τους να είναι 0,002 ns που αντιστοιχεί σε ποσοστό 1,16%. Ενδεικτικά οι τιμές κυμαίνονται από 0,02 ns έως 2,7 ns.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,01 ns έως 1,1 ns, ενώ στο σημείο SS από 0,04 ns έως 1,91 ns. Όπως δείχνει και το σχήμα 2.86, η μέση διαφορά μεταξύ τους είναι 0,20 ns που αντιστοιχεί σε 50,76% και η μέγιστη διαφορά είναι 0,88 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	2400	-0.205572	-50.76%	9.65%	-0.879923	-114.44%	2400

Σχήμα 2.86: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί OAl22_X0d5.



Σχήμα 2.87: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί OAl22_X0d5.

Στο σχήμα 2.88 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,002 ns που αντιστοιχεί σε ποσοστό 0,7% με ταχύτερο το τροποποιημένο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,13 ns έως 4,1 ns για την άνοδο και από 0,03 ns έως 1,83 ns για την κάθοδο.



Σχήμα 2.88: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί OAI22_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,08 ns έως 1,25 ns για την άνοδο και από 0,025 ns έως 0,85 ns για την κάθοδο. Ενώ στο σημείο SS από 0,19 ns έως 2,49 ns για την άνοδο και από 0,045 ns έως 1,16 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.89, η μέση διαφορά μεταξύ τους είναι 0,282 ns που αντιστοιχεί σε 44,3% και η μέγιστη διαφορά είναι 1,4 ns.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
 trans(ns)	2400	-0.281966	-44.29%	14.16%	-1.399342	-61,19%	2400

Σχήμα 2.89: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί OAl22_X0d5.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.90. Το τροποποιημένο κελί έχει ελάχιστα μεγαλύτερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,0002 pJ που αντιστοιχεί σε 7,5%. Ενδεικτικά οι τιμές κυμαίνονται από 0,001 pJ έως 0,014 pJ.



Σχήμα 2.90: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί OAI22_X0d5.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,001 pJ έως 0,038 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,017 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.83, η μέση διαφορά μεταξύ τους είναι 0,003 pJ που αντιστοιχεί σε 29,17%.

	OAI22_X0d5	power	SUMMARY	(abstol	0.001pJ,	reltol	1%)
--	------------	-------	---------	---------	----------	--------	-----

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	2800	0.002985	29.17%	53,39%	0.020820 1	308.21%	1519
Worst	power outlier	: Max Abs:	0.020820, F	Row # : 1238;	Max Rel:	308.21%, Row # :	122	



2.7.10 Αποτελέσματα για το κελί DFC_X1

Στο σχήμα 2.92 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 104,24 pW έναντι 77,15 pW της πρωτότυπης και είναι αυξημένη κατά 27,08 pW, δηλαδή 35,3%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	leakage(pW)	14	27.086239	35,28%	4.85%	34.561113	45.18%	14

Σχήμα 2.92: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X1.

Στο σχήμα 2.93 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 93.8 pW για σημείο FF, έναντι 112,45 pW για SS, δηλαδή στο σημείο FF έχουμε χαμηλότερη κατανάλωση. Υπάρχει μια διαφορά 18,65 pW που είναι της τάξης του 16,44%.

Data 1	rype	Entries	Avg Diff	Avg Diff%	Sigma≹	Max Diff	Max Diff%	Outliers
leakage	(pW)	14	-18.658200	-16.44%	1.59%	-24.982000	-19.22%	14

Σχήμα 2.93: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί DFC_X1.

Στο σχήμα 2.94 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Το τροποποιημένο κελί είναι ταχύτερο, με την μέση διαφορά μεταξύ τους να είναι 0,003 ns που αντιστοιχεί σε ποσοστό 6,9%. Ενδεικτικά οι τιμές κυμαίνονται από 0,11 ns έως 1,53 ns.



Σχήμα 2.94: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,08 ns έως 1,1 ns, ενώ στο σημείο SS από 0,16 ns έως 2,70 ns. Όπως δείχνει και το σχήμα 2.95, η μέση διαφορά μεταξύ τους είναι 0,33 ns που αντιστοιχεί σε 49,79% και η μέγιστη διαφορά είναι 1,56 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma)	Max Diff	Max Diff%	Outliers
1	delay(ns) (600	-0.334015	-49.79%	5.54%	-1.564680	-58.03%	600

Σχήμα 2.95: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί DFC_X1.

Στο σχήμα 2.96 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,002 ns που αντιστοιχεί σε ποσοστό 0,5% με απειροελάχιστα ταχύτερο το πρωτότυπο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,04 ns έως 1,87 ns για την άνοδο και από 0,03 ns έως 0,91 ns για την κάθοδο.



Σχήμα 2.96: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,02 ns έως 1,8 ns για την άνοδο και από 0,002 ns έως 1,03 ns για την κάθοδο. Ενώ στο σημείο SS από 0,02 ns έως 4,3 ns για την άνοδο και από 0,04 ns έως 1,87 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.97, η μέση διαφορά μεταξύ τους είναι 0,29 ns που αντιστοιχεί σε 50,7% και η μέγιστη διαφορά είναι 2,5 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff*	Outliers
1	trans(ns)	600	-0.291350	-50.75%	5.80%	-2.538130	-58.07%	600

Σχήμα 2.97: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί DFC_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.98. Το τροποποιημένο κελί έχει ελάχιστα μικρότερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,003 pJ που αντιστοιχεί σε 31,1%. Ενδεικτικά οι τιμές κυμαίνονται από 0,012 pJ έως 0,057 pJ.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,001 pJ έως 0,069 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,05 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.99, η μέση διαφορά μεταξύ τους είναι 0,006 pJ που αντιστοιχεί σε 18,31%.



Σχήμα 2.98: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X1.

+	AI DOWEL SOMMAP	(I (ADSCOI U.	colps, reitor	14)				++
Ľ	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	760	0.006098	18.314	1 20.334	0.030889	206.56%	680
Wors	t power outlier	: Max Abs:	0.030889, 1	Row # :	50; Max Rel:	206.56%, Row # :	300	++

Σχήμα 2.99: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί DFC_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για τους χρονικούς περιορισμούς για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις είναι ταχύτερο το ένα κελί και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.100. Το πρωτότυπο κελί είναι ταχύτερο με την μέση διαφορά μεταξύ τους να είναι 0,07 ns που αντιστοιχεί σε 36,1%. Ενδεικτικά οι τιμές κυμαίνονται από -0,04 ns έως 0,46 ns για τον χρόνο προετοιμασίας, από -0,29 ns έως 1,01 ns για τον χρόνο συγκράτησης, από 0,17 ns έως 0,61 ns για τον χρόνο απομάκρυνσης και από -0,25 ns έως 0,41 ns για τον χρόνο αποκατάστασης.



Σχήμα 2.100: Διάγραμμα σύγκρισης χρονικών περιορισμών μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFC_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών των χρονικών περιορισμών είναι στο σημείο FF από -0,18 ns έως 0,39 ns για τον χρόνο προετοιμασίας, από 0,28 ns έως 0,06 ns για τον χρόνο συγκράτησης, από 0,12 ns έως 0,48 ns για τον χρόνο απομάκρυνσης και από -0,15 ns έως 0,39 ns για τον χρόνο αποκατάστασης. Στο σημείο SS είναι από -0,16 ns έως 0,59 ns για τον χρόνο προετοιμασίας, από 0,32 ns έως 0,249 ns για τον χρόνο συγκράτησης, από 0,24 ns έως 0,88 ns για τον χρόνο απομάκρυνσης και από -0,48 ns έως 0,39 ns για τον χρόνο αποκατάστασης. Η σημείο FF είναι λίγο ταχύτερη σε σχέση με την SS και όπως δείχνει και το σχήμα 2.101, η μέση διαφορά μεταξύ τους είναι 0,03 ns που αντιστοιχεί σε 3,73%.

DFC_X1 constraint	: SUMMARY (a	bstol 0.001ns, rel	tol 1%)	¥			
Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
constraint(ns)	1 165	-0.036747	3.73%	236.20%	-0.398429	2521.11%	156
Worst constraint	outlier: Ma	x Abs: -0.39842	9, Row # :	53; Max Rel:	2521.11%,	Row # :	++ 151

Σχήμα 2.101: Σύγκριση χρονικών περιορισμών μεταξύ σημείων FF και SS για το κελί DFC_X1.

2.7.11 Αποτελέσματα για το κελί DFNC_X1

Στο σχήμα 2.102 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 95,38 pW έναντι 70,39 pW της πρωτότυπης και είναι αυξημένη κατά 24,98 pW, δηλαδή 35,8%.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	leakage(pW)	14	24,986249	35.78%	6.17%	32.430397	46.85%	14

Σχήμα 2.102: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1.

Στο σχήμα 2.103 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 86.09 pW για σημείο FF, έναντι 102,78 pW για SS, δηλαδή στο σημείο FF έχουμε χαμηλότερη κατανάλωση. Υπάρχει μια διαφορά 16,7 pW που είναι της τάξης του 16,17%.

I	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
l lea	kage (nW)	14	-16,694314 (-16.17%	0.89%	-21,657000 1	-17.74%	14

Σχήμα 2.103: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί DFNC_X1.

Στο σχήμα 2.104 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Το τροποποιημένο κελί είναι ελαφρώς ταχύτερο, με την μέση διαφορά μεταξύ τους να είναι 0,021 ns που αντιστοιχεί σε ποσοστό 5,13%. Ενδεικτικά οι τιμές κυμαίνονται από 0,1 ns έως 1,5 ns.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,66 ns έως 1,24 ns, ενώ στο σημείο SS από 0,14 ns έως 2,75 ns. Όπως δείχνει και το σχήμα 2.105, η μέση διαφορά μεταξύ τους είναι 0,3 ns που αντιστοιχεί σε 48,29% και η μέγιστη διαφορά είναι 1,51 ns.



Σχήμα 2.104: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	600	-0.313534 /	-48.29%	5.19%	-1.515570	-57.74%	600

Σχήμα 2.105: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί DFNC_X1.

Στο σχήμα 2.106 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,002 ns που αντιστοιχεί σε ποσοστό 0,08% με απειροελάχιστα ταχύτερο το πρωτότυπο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,04 ns έως 1,87 ns για την άνοδο και από 0,03 ns έως 0,90 ns για την κάθοδο.



Σχήμα 2.106: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,03 ns έως 1,83 ns για την άνοδο και από 0,023 ns έως 1,01 ns για την κάθοδο. Ενώ στο σημείο SS από 0,06 ns έως 4,37 ns για την άνοδο και από 0,046 ns έως 1,85 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.107, η μέση διαφορά μεταξύ τους είναι 0,29 ns που αντιστοιχεί σε 50,6% και η μέγιστη διαφορά είναι 2,53 ns.

1	Data Type	Entries	Avg Diff	Avg Diff	k∣ Sigma%	Max Diff	Max Diff%	Outliers
1	trans(ns)	600	-0.290461	-50,644	1 5.98%	-2.539750	-58.10%	600

Σχήμα 2.107: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί DFNC_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.108. Το τροποποιημένο κελί έχει ελάχιστα μικρότερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,002 pJ που αντιστοιχεί σε 1,1%. Ενδεικτικά οι τιμές κυμαίνονται από 0,001 pJ έως 0,047 pJ.



Σχήμα 2.108: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,001 pJ έως 0,07 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,044 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.109, η μέση διαφορά μεταξύ τους είναι 0,006 pJ που αντιστοιχεί σε 22,36%.

	Por 1	putties	Avg Diff	Avg Diffs	Sigma%	Max Diff	Max Diff%	Outliers
power(J) 1	760	0.006470	22.36%	1 36.36%	0.033262	465.20%	673

Σχήμα 2.109: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί DFNC_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για τους χρονικούς περιορισμούς για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις είναι ταχύτερο το ένα κελί και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.110. Το πρωτότυπο κελί είναι ταχύτερο με την μέση διαφορά μεταξύ τους να είναι 0,06 ns που αντιστοιχεί σε 20,22%. Ενδεικτικά οι τιμές κυμαίνονται από -0,07 ns έως 0,49 ns για τον χρόνο προετοιμασίας, από -0,29 ns έως 0,97 ns για τον χρόνο συγκράτησης, από 0,12 ns έως 0,68 ns για τον χρόνο απομάκρυνσης και από -0,34 ns έως 0,43 ns για τον χρόνο αποκατάστασης.



Σχήμα 2.110: Διάγραμμα σύγκρισης χρονικών περιορισμών μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFNC_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών των χρονικών περιορισμών είναι στο σημείο FF από -0,18 ns έως 0,41 ns για τον χρόνο προετοιμασίας, από -0,28 ns έως 0,18 ns για τον χρόνο συγκράτησης, από 0,085 ns έως 0,84 ns για τον χρόνο απομάκρυνσης και από -0,44 ns έως 0,41 ns για τον χρόνο αποκατάστασης. Στο σημείο SS είναι από -0,19 ns έως 0,62 ns για τον χρόνο προετοιμασίας, από -0,32 ns έως 0,218 ns για τον χρόνο συγκράτησης, από 0,18 ns έως 1,01 ns για τον χρόνο απομάκρυνσης και από -0,64 ns έως 0,42 ns για τον χρόνο αποκατάστασης. Η σημείο FF είναι λίγο ταχύτερη σε σχέση με την SS και όπως δείχνει και το σχήμα 2.111, η μέση διαφορά μεταξύ τους είναι 0,033 ns που αντιστοιχεί σε 34,26%.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma 🕴	Max Diff	Max Diff*	Outliers
constraint(ns)	165	-0.033182	-34.26%	259.54%	-0,331580	-2562.64%	156

Σχήμα 2.111: Σύγκριση χρονικών περιορισμών μεταξύ σημείων FF και SS για το κελί DFNC_X1.

2.7.12 Αποτελέσματα για το κελί DFCS_X1

Στο σχήμα 2.112 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την στατική ισχύ, της οποίας η μέση τιμή είναι 118,24 pW έναντι 86,33 pW της πρωτότυπης και είναι αυξημένη κατά 31,91 pW, δηλαδή 36,94%.

I	Data Type	Entries	Avg Diff	Avg Diff% (Sigma%	Max Diff	Max Diff%	Outliers
1 1	eakage(pW)	50 1	31.913693	36.94% 1	2.95% 1	40.067462	43.09%	50

Σχήμα 2.112: Σύγκριση στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1.

Στο σχήμα 2.113 βλέπουμε την στατική ισχύ για σημείο FF στους -40 C° και σημείο SS στους 70 C°. Η μέση ισχύς είναι 106.8 pW για σημείο FF, έναντι 127,23 pW για SS, δηλαδή στο σημείο FF έχουμε χαμηλότερη κατανάλωση. Υπάρχει μια διαφορά 20,45 pW που είναι της τάξης του 16%.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
leakage(pW)	50	-20.451704	-15.99%	1.33%	-26.242000	-18.51%	50

Σχήμα 2.113: Σύγκριση στατικής ισχύος μεταξύ σημείων FF και SS για το κελί DFCS_X1.

Στο σχήμα 2.114 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για την καθυστέρηση διάδοσης για σημείο TT. Το τροποποιημένο κελί είναι ελαφρώς ταχύτερο, με την μέση διαφορά μεταξύ τους να είναι 0,02 ns που αντιστοιχεί σε ποσοστό 4,44%. Ενδεικτικά οι τιμές κυμαίνονται από 0,1 ns έως 1,5 ns.



Σχήμα 2.114: Γράφημα για τη σύγκριση καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της καθυστέρησης είναι στο σημείο FF από 0,07 ns έως 1,13 ns, ενώ στο σημείο SS από 0,15 ns έως 2,71 ns. Όπως δείχνει και το σχήμα 2.115, η μέση διαφορά μεταξύ τους είναι 0,3 ns που αντιστοιχεί σε 49,04% και η μέγιστη διαφορά είναι 1,57 ns.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma% (Max Diff	Max Diff%	Outliers
1	delay(ns)	1800	-0.294575	-49.04%	5.13%	-1.572430	-59.50%	1800

Σχήμα 2.115: Σύγκριση καθυστέρησης μεταξύ σημείων FF και SS για το κελί DFCS_X1.

Στο σχήμα 2.116 βλέπουμε τη σύγκριση της βιβλιοθήκης μας με την πρωτότυπη για τον χρόνο μετάβασης της εξόδου για σημείο TT. Τα δύο κελιά είναι σχεδόν ίδια με την μέση διαφορά μεταξύ τους να είναι στα 0,003 ns που αντιστοιχεί σε ποσοστό 0,10% με απειροελάχιστα ταχύτερο το πρωτότυπο κελί. Ενδεικτικά οι τιμές κυμαίνονται από 0,04 ns έως 1,87 ns για την άνοδο και από 0,03 ns έως 0,91 ns για την κάθοδο.



Σχήμα 2.116: Γράφημα σύγκρισης χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών του χρόνου μετάβασης της εξόδου είναι στο σημείο FF από 0,03 ns έως 1,83 ns για την άνοδο και από 0,023 ns έως 1,03 ns για την κάθοδο. Ενώ στο σημείο SS από 0,059 ns έως 4,37 ns για την άνοδο και από 0,045 ns έως 1,87 ns για την κάθοδο. Όπως δείχνει και το σχήμα 2.117, η μέση διαφορά μεταξύ τους είναι 0,25 ns που αντιστοιχεί σε 50,25% και η μέγιστη διαφορά είναι 2,53 ns.

*	Data Type	t Entrice	Awa Diff	Awa Diffs	sima%	Max Diff	Max Diffà Outli	ers I
	buon Tipo (nty batt 1			THE DALL	Hun Dilli Outin	
1	trans(ns)	1800	-0.250623	-50.25%	6.16%	-2.538700	-58.07% 1	800
+ Worst	trans outlier:	Max Abs:	-2.538700, R	ow # : 850; Max	Rel:	-58.07%, Row # :	850	+

Σχήμα 2.117: Σύγκριση χρόνου μετάβασης εξόδου μεταξύ σημείων FF και SS για το κελί DFCS_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για την κατανάλωση ενέργειας για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις έχει παραπάνω κατανάλωση το ένα και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.118. Το τροποποιημένο κελί έχει ελάχιστα μικρότερη μέση κατανάλωση, με την μέση διαφορά μεταξύ τους να είναι 0,002 pJ που αντιστοιχεί σε 36,73%. Ενδεικτικά οι τιμές κυμαίνονται από 0,001 pJ έως 0,06 pJ.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών της κατανάλωσης ενέργειας είναι στο σημείο FF από 0,001 pJ έως 0,085 pJ, ενώ στο σημείο SS από 0,001 pJ έως 0,051 pJ. Η σημείο FF έχει μεγαλύτερη κατανάλωση ενέργειας σε σχέση με την SS. Όπως δείχνει και το σχήμα 2.119, η μέση διαφορά μεταξύ τους είναι 0,005 pJ που αντιστοιχεί σε 23,81%.



Σχήμα 2.118: Διάγραμμα σύγκρισης κατανάλωσης ενέργειας μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ) (2920 1	0.005657	23.81% (96.05%	0.045578 (1632.51%	2310

Σχήμα 2.119: Σύγκριση κατανάλωσης ενέργειας μεταξύ σημείων FF και SS για το κελί DFCS_X1.

Στη σύγκριση του τροποποιημένου κελιού με το πρωτότυπο για τους χρονικούς περιορισμούς για σημείο TT, παρατηρούμε ότι σε άλλες περιπτώσεις είναι ταχύτερο το ένα κελί και σε άλλες το άλλο, όπως βλέπουμε στο σχήμα 2.120. Το πρωτότυπο κελί είναι ταχύτερο με την μέση διαφορά μεταξύ τους να είναι 0,042 ns που αντιστοιχεί σε 14,29%. Ενδεικτικά οι τιμές κυμαίνονται από -0,08 ns έως 0,60 ns για τον χρόνο προετοιμασίας, από -0,020 ns έως -0,41 ns για τον χρόνο συγκράτησης, από 0,12 ns έως 0,49 ns για τον χρόνο απομάκρυνσης και από -0,20 ns έως 0,50 ns για τον χρόνο αποκατάστασης.



Σχήμα 2.120: Διάγραμμα σύγκρισης χρονικών περιορισμών μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης για το κελί DFCS_X1.

Για σημείο FF στους -40 C° και σημείο SS στους 70 C° το εύρος τιμών των χρονικών περιορισμών είναι στο σημείο FF από 0,07 ns έως 0,47 ns για τον χρόνο προετοιμασίας, από -0,02 ns έως -0,35 ns για τον χρόνο συγκράτησης, από 0,087 ns έως 0,40 ns για τον χρόνο απομάκρυνσης και από -0,123 ns έως 0,47 ns για τον χρόνο αποκατάστασης. Στο σημείο SS είναι από 0,066 ns έως 0,88 ns για τον χρόνο προετοιμασίας, από -0,57 ns έως 0,044 ns για τον χρόνο συγκράτησης, από 0,17 ns έως 0,71 ns για τον χρόνο απομάκρυνσης και από -0,55 ns για τον χρόνο αποκατάστασης. Η σημείο FF είναι λίγο ταχύτερη σε σχέση με την SS και όπως δείχνει και το σχήμα 2.121, η μέση διαφορά μεταξύ τους είναι 0,044 ns που αντιστοιχεί σε 17,47%.

DFCS_X1 constraint SUMMARY (abstol 0.001ns, reltol 1%)

+	+		*-				**
I Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
constraint(ns)	365	-0.044272	-17.47%	116.92%	-0.435055	-1305.71%	359
Worst constraint	outlier: Max	Abs: -0.43505	55, Row # :	355; Max Rel:	-1305.71%,	Row # :	99

Σχήμα 2.121: Σύγκριση χρονικών περιορισμών μεταξύ σημείων FF και SS για το κελί DFCS_X1.

3. ΥΛΟΠΟΙΗΣΗ ΤΟΥ SPI SLAVE

Στο κεφάλαιο αυτό αρχικά θα δούμε τι είναι το πρωτόκολλο επικοινωνίας SPI και η μέθοδος σχεδίασης με χρήση τριπλού πλεονασμού (TMR). Στη συνέχεια θα δούμε λεπτομέρειες για τη σύνθεση και τη φυσική σχεδίαση δύο υλοποιήσεων του slave του SPI. Μία υλοποίηση με χρήση απλού TMR, δηλαδή τριπλασιασμό μόνο των καταχωρητών και μία υλοποίηση με χρήση πλήρους TMR, δηλαδή τριπλασιασμό καταχωρητών, συνδυαστικής λογικής, εισόδων και εξόδων. Τέλος θα δούμε τις προσομοιώσεις και τα αποτελέσματα της κάθε υλοποίησης και θα γίνει σύγκριση των αποτελεσμάτων των δύο υλοποιήσεων, καθώς και σύγκριση με χρήση δακτυλίων προστασίας και χρήση κανονικών κελιών.

3.1 Πρωτόκολλο SPI [40]

Το πρωτόκολλο επικοινωνίας SPI, είναι ένα πρωτόκολλο σειριακής μεταφοράς δεδομένων μεταξύ κύριου (master) και υποτελή (slave) χρησιμοποιώντας τέσσερα σήματα. Ο κύριος (master) ξεκινά την ανταλλαγή δεδομένων με τον slave, κάνοντας το σήμα SS (Select Slave) '0'. Το σήμα σειριακού ρολογιού (SCLK), που οδηγείται από τον master, παρέχει μία πηγή σύγχρονου ρολογιού. Ο master μεταδίδει δεδομένα μέσω του σήματος MOSI (Master Out, Slave In) και λαμβάνει δεδομένα μέσω του σήματος MISO (Master In, Slave Out). Ο master μπορεί να επικοινωνεί με πολλαπλούς slaves μέσω μιας ποικιλίας τεχνικών. Στην πιο συνηθισμένη διάταξη, κάθε slave έχει μια ανεξάρτητη γραμμή SS αλλά μοιράζεται τις γραμμές SCLK, MISO και MOSI με τους άλλους slaves. Κάθε slave αγνοεί τις κοινές γραμμές όταν η γραμμή SS είναι στην λογική κατάσταση '1'. Μια τέτοια διάταξη απεικονίζεται στο σχήμα 3.1.



Σχήμα 3.1: SPI διάταξη με τρείς slaves. [41]

Ο SPI έχει τέσσερις τρόπους λειτουργίας που βασίζονται σε δύο παραμέτρους: την πολικότητα του ρολογιού (CPOL) και τη φάση του ρολογιού (CPHA). O master και ο slave πρέπει να χρησιμοποιούν τον ίδιο τρόπο λειτουργίας για να επικοινωνούν σωστά. Αν το CPOL είναι '0', τότε το SCLK είναι αρχικά σε λογική κατάσταση '0' και η πρώτη αιχμή του ρολογιού είναι ανοδική. Αν το CPOL είναι '1', το SCLK είναι κανονικά στη λογική κατάσταση '1' και η πρώτη αιχμή του ρολογιού είναι καθοδική. Το CPHA ορίζει την ευθυγράμμιση δεδομένων. Εάν το CPHA είναι '0', τότε το πρώτο bit δεδομένων γράφεται στην καθοδική αιχμή του SS και διαβάζεται στην πρώτη αιχμή του SCLK. Εάν το CPHA είναι '1', τα δεδομένα γράφονται στην πρώτη αιχμή του SCLK και διαβάζονται στη δεύτερη αιχμή του SCLK. Το διάγραμμα χρονισμού στο σχήμα 3.2 απεικονίζει τους τέσσερις τρόπους λειτουργίας του SPI.

00114	CPOL = 0							1		_	1					_	
SULK	CPOL = 1	-											5		1		
SS		\vdash		_	_	_	_				_	-	-	_	_		
MOSI/	CPHA = 0	zx	1 X	2	×	3	×	4	×	5	×	6	×	7	×	8	XXXZ
MISO	CPHA = 1	ZXX	XI		2	×	3	×	4	×	5	×	6	×	7	×	8 XZ

Σχήμα 3.2: Διάγραμμα χρονισμών με τους τρόπους λειτουργίας του SPI. [40]

Η χρονική διάρκεια κάθε συναλλαγής μεταξύ master και slave εξαρτάται από τα πόσα bit είναι τα δεδομένα που θα μεταφερθούν συν το bit που διευκρινίζει αν πρόκειται για ανάγνωση ή εγγραφή. Για τις ανάγκες της εργασίας, η επικοινωνία του master με τον slave απότελείται από 13 bits, δηλαδή 13 κύκλους ρολογιού. Η σημασία των bits αυτών θα εξηγηθεί στην ενότητα 3.3.

3.2 Λειτουργία τριπλού πλεονασμού (TMR) [42]

Μία γνωστή μέθοδος για την μείωση των διαταράξεων απλού συμβάντος (SEUs) είναι η λειτουργία τριπλού πλεονασμού (Triple Mode Redundancy - TMR) με ψηφοφορία. Αυτή η μέθοδος μείωσης χρησιμοποιεί τρία πανομοιότυπα λογικά κυκλώματα που εκτελούν την ίδια λειτουργία παράλληλα, με τις εξόδους του να συγκρίνονται μέσω ενός κυκλώματος ψηφοφόρων πλειοψηφίας. Το πιο κοινό παράδειγμα του TMR είναι ένα flipflop τύπου D που έχει τριπλασιαστεί και στο οποίο έχει προστεθεί στην έξοδο του το κύκλωμα ψηφοφορίας. Με την αντικατάσταση όλων των flip-flops με το κύκλωμα που φαίνεται στο σχήμα 3.3, θα επιτευχθεί ανθεκτικότητα από τις διαταράξεις απλού συμβάντος στα flip flops. Ωστόσο, αυτό δεν θα προστατεύσει το κύκλωμα συνδυαστικής λογικής που συνδέει τα flip flops.



Σχήμα 3.3: Λειτουργία τριπλού πλεονασμού με ψηφοφορία. [42]

Οι επιπτώσεις των SEUs δεν περιορίζονται στους καταχωρητές στα ψηφιακά κυκλώματα, αλλά επίσης μπορούν να παρουσιαστούν στη συνδυαστική λογική. Αυτό απαιτεί μια τεχνική προστασίας που να καλύπτει πλήρως το κύκλωμα, τόσο το ακολουθιακό, όσο και το συνδυαστικό τμήμα. Μια πολύ απλή μέθοδος για την μείωση των SEU σε ένα κύκλωμα, είναι η δημιουργία αντιγράφων ολόκληρων μονάδων και με χρήση ψηφοφόρων που τοποθετούνται στις εξόδους τους, να ψηφίσουν τα τελικά αποτελέσματα. Σε αυτήν την περίπτωση μία μονάδα μπορεί να αντιπροσωπεύει είτε το σύνολο του κυκλώματος, είτε ένα υποσύστημα του κυκλώματος. Πρόκειται για ένα πολύ αποτελεσματικό τρόπο μείωσης της SEU, ο οποίος είναι εύκολο να εφαρμοστεί. Το σχέδιο μίας τέτοιας διάταξης απεικονίζεται στο σχήμα 3.4.



Σχήμα 3.4: Τριπλός πλεονασμός υποσυστημάτων. [42]

Το μειονέκτημα της τεχνικής σε επίπεδο υποσυστημάτων είναι ότι δεν παρέχει έναν απλό και ισχυρό μηχανισμό διόρθωσης μετά από τον εντοπισμό ενός σφάλματος σε ένα από τα υποσυστήματα. Σε περίπτωση σφάλματος σε ένα υποσύστημα, αυτό μπορεί να διατηρηθεί και το συγκεκριμένο υποσύστημα να βγάζει συνεχώς εσφαλμένα δεδομένα στην έξοδο του. Σε αυτή την περίπτωση αν προκληθεί κάποιο άλλο σφάλμα σε κάποιο από τα αντίγραφα, τότε θα έχουμε λανθασμένο αποτέλεσμα και στην έξοδο των ψηφοφόρων και σαν αποτέλεσμα θα υπάρχει απώλεια δεδομένων.

Μία άλλη μέθοδος για την προστασία των ψηφιακών κυκλωμάτων από τις SEUs, είναι η χρήση τριπλού πλεονασμού σε επίπεδο πύλης. Η ανατροφοδότηση του αποτελέσματος των ψηφοφόρων είναι σημαντική για να αποκατασταθεί η κατάσταση όλων των πλεοναζόντων υποσυστημάτων και να αποφευχθεί η δημιουργία σφαλμάτων. Το πλεονέκτημα της τεχνικής σε επίπεδο πύλης, είναι ότι η ψηφοφορία μπορεί να εκτελεστεί μετά τα ακολουθιακά στοιχεία, πριν, ή και στο εσωτερικό των συνδυαστικών κυκλωμάτων, ανάλογα με το επίπεδο προστασίας που απαιτείται. Το αποτέλεσμα που έχει ψηφιστεί συνήθως τροφοδοτείται στο συνδυαστικό τμήμα και έτσι σε περίπτωση σφάλματος αυτό διορθώνεται, αποφεύγοντας έτσι ότι το σφάλμα θα διαδοθεί στο ακολουθιακό τμήμα και από εκεί στο υπόλοιπο κύκλωμα. Αυτή η διόρθωση πραγματοποιείται σε ένα κύκλο ρολογιού. Στο σχήμα 3.5 απεικονίζεται ένα υποσύστημα τριπλού πλεονασμού σε επίπεδο πύλης, το οποίο έχει τριπλό πλεονασμό και στο συνδυαστικό και στο ακολουθιακό κομμάτι και στους ψηφοφόρους, οι οποίοι ψηφοφόροι χρησιμοποιούνται και στο συνδυαστικό και στο ακολουθιακό τμήμα. Όπως αναφέρθηκε και παραπάνω, η έξοδος των ψηφοφόρων που συνοδεύουν την ακολουθιακή λογική, ανατροφοδοτούν τη συνδυαστική λογική, διορθώνοντας το όποιο σφάλμα μπορεί να προκύψει.



Σχήμα 3.5: Υποσύστημα με χρήση τριπλού πλεονασμού σε επίπεδο πύλης. [42]

3.3 Υλοποίηση του SPI slave με χρήση τριπλού πλεονασμού

Στην ενότητα αυτή θα δούμε δύο διαφορετικές υλοποιήσεις του SPI slave, οι οποίες θα είναι και οι δύο με χρήση τριπλού πλεονασμού και θα γίνουν σύνθεση με τα κελιά που αναφέρθηκαν στο περασμένο κεφάλαιο. Η μία υλοποίηση θα είναι με τριπλασιασμό μόνο

των καταχωρητών και χρήση ενός ψηφοφόρου (voter) στην έξοδο των καταχωρητών (απλή TMR υλοποίηση) και η δεύτερη υλοποίηση θα είναι με τριπλασιασμό και του συνδυαστικού μέρους και του ακολουθιακού μέρους και των ψηφοφόρων (voters) και των εισόδων και εξόδων (πλήρης TMR υλοποίηση). Σε αυτή την εργασία θα γίνει σύνθεση και φυσική σχεδίαση μόνο του slave (υποτελής) και όχι του master (κύριος). Για τον master θα γίνει περιγραφή της λειτουργίας του μόνο σε επίπεδο προσομοίωσης. Επίσης για τις ανάγκες της προσομοίωσης και μόνο, θα υλοποιηθεί και μία μνήμη 16 θέσεων των 8-bit για την εγγραφή και ανάγνωση των αποτελεσμάτων.

3.3.1 Προδιαγραφές

Για τις ανάγκες της εργασίας ο slave θα χρησιμοποιείται για την εγγραφή και ανάγνωση δεδομένων από μία μνήμη και τα οποία θα τα ανταλλάζει σειριακά με τον master. Επομένως θα υπάρχουν μόνο οι κατάλληλες είσοδοι και έξοδοι για την επικοινωνία με αυτές τις δύο μονάδες. Αυτές οι θύρες είναι οι εξής:

Είσοδοι

- i_SCLK: Είναι το σήμα ρολογιού που παίρνει από τον master και με αυτό συγχρονίζονται όλες οι υπομονάδες του slave.
- i_rstn: Σήμα επαναφοράς (reset). Ενεργοποιείται όταν είναι '0'.
- i_CSN: Σήμα ενεργοποίησης του slave. Προέρχεται από τον master και ο slave είναι ενεργός όταν το σήμα είναι στην λογική κατάσταση '0'.
- i_MOSI: Σειριακή είσοδος για την μεταφορά δεδομένων από τον master. (Master Out Slave In).
- i_data_in: Δίαυλος μήκους 8-bit, για τη λήψη δεδομένων από τη μνήμη κατά τη λειτουργία της ανάγνωσης.

Έξοδοι:

- o_wr_en: Σήμα ενεργοποίησης της εγγραφής στη μνήμη (Write enable). Όταν είναι στην λογική κατάσταση '1' γράφει στη μνήμη τα περιεχόμενα της θύρας o_data_out στην θέση μνήμης που υποδηλώνει η θύρα o_addr_out..
- o_MISO: Σειριακή έξοδος για την μεταφορά δεδομένων προς τον master. (Master In – Slave out).
- o_addr_out: Δίαυλος μήκους 4-bit που δηλώνει την διεύθυνση της μνήμης που θα γίνει προσπέλαση.
- o_data_out: Δίαυλος μήκους 8-bit, για την αποστολή δεδομένων προς τη μνήμη κατά τη λειτουργία της εγγραφής.

Όταν το σήμα ενεργοποίησης του slave (i_CSN) γίνει '0', τότε ξεκινάει η επικοινωνία μεταξύ master και slave. Αυτή η επικοινωνία έχει διάρκεια 13 κύκλους ρολογιού. Από αυτούς τους 13 κύκλους, στους 4 πρώτους μεταφέρονται σειριακά τα 4 bits που είναι η διεύθυνση της μνήμης που θα χρησιμοποιηθεί, δηλαδή μία από τις 16 θέσεις της μνήμης (2⁴=16). Το 5° bit είναι αυτό που καθορίζει αν θα γίνει εγγραφή σε αυτή τη διεύθυνση ή ανάγνωση από αυτή. Αυτά τα 5 bits στέλνονται από τον master στον slave μέσω της θύρας i_MOSI. Τέλος οι τελευταίοι 8 κύκλοι είναι τα 8 bits των δεδομένων.

Κατά τη λειτουργία εγγραφής ο master θα στέλνει σειριακά μέσω της θύρας i_MOSI τα 8 bit που θα εγγραφούν στη μνήμη, στη διεύθυνση που ορίστηκε στους 4 πρώτους κύκλους ρολογιού της μεταξύ τους επικοινωνίας. Στον τελευταίο κύκλο η έξοδος ενεργοποίησης της εγγραφής (o_wr_en) θα παίρνει την τιμή ´1´ και θα ενεργοποιείται η μνήμη για να αποθηκεύσει τα δεδομένα μέσω του διαύλου o_data_out, στην θέση μνήμης που ορίζεται

μέσω του διαύλου διεύθυνσης o_addr_out. Ο slave θα πρέπει να δουλεύει σε λειτουργία με CPOL = '0' και CPHA = '0'.

Κατά τη λειτουργία ανάγνωσης ο slave θα στέλνει σειριακά μέσω της θύρας MISO τα 8 bit που αναγνώστηκαν από τη μνήμη, στη διεύθυνση που ορίστηκε στους 4 πρώτους κύκλους ρολογιού της μεταξύ τους επικοινωνίας.

Η διεύθυνση και τα δεδομένα μεταξύ master και slave στέλνονται όλα ξεκινώντας με το υψηλότερης σημασίας bit (bit 7). Τέλος η ανάγνωση και η εγγραφή στη μνήμη θα γίνονται παράλληλα, σε ένα κύκλο ρολογιού.

3.3.2 Σύνθεση του slave

Παρακάτω θα δούμε τις υπομονάδες με τις οποίες θα υλοποιήσουμε τον slave του SPI. Η σύνθεση πραγματοποιήθηκε με το εργαλείο της Cadence, το Genus την έκδοση v15.20.

3.3.2.1 Ψηφοφόρος (Voter)

Ο ψηφοφόρος είναι ένα συνδυαστικό κύκλωμα που αποτελείται από τρείς εισόδους του ενός bit και μία έξοδο του ενός bit επίσης. Αυτό που κάνει ο ψηφοφόρος είναι να περνάει στην έξοδο την τιμή που υπερισχύει στις εισόδους, δηλαδή την τιμή που είναι σε τουλάχιστον δύο εισόδους. Ο πίνακας αληθείας αυτού του κυκλώματος απεικονίζεται στον πίνακα 1.

A	В	C	v
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Πίνακας 1: Πίνακας αληθείας ψηφοφόρου.

Η λογική συνάρτηση που προκύπτει από τον πίνακα αληθείας είναι V= AB + AC + BC και το λογικό κύκλωμα απεικονίζεται στο σχήμα 3.6.



Σχήμα 3.6: Λογικό κύκλωμα ψηφοφόρου.

Ο ψηφοφόρος συνδέεται στις εξόδους των πλεοναζόντων κυκλωμάτων(ακολουθιακά ή συνδυαστικά). Σε περίπτωση που οι έξοδοι των πλεοναζόντων κυκλωμάτων είναι παραπάνω από 1 bit, τότε για κάθε bit αντιστοιχεί και ένας ψηφοφόρος, όπου ο κάθε ψηφοφόρος συγκρίνει τα bit ίδιας σημασίας των εξόδων. Το σχηματικό που προέκυψε μετά την σύνθεση, απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 3.7: Σχηματικό ψηφοφόρου μετά την σύνθεση.

Παρατηρούμε ότι κατά την σύνθεση επιλέχτηκαν δύο πύλες NOR, δύο αντιστροφείς και ένα κελί AO21, αντί γι αυτό του σχήματος 3.6. Αυτό συνέβη διότι η βιβλιοθήκη μας δεν περιλαμβάνει πύλη OR δύο ή τριών εισόδων, οπότε επιλέχτηκε η βέλτιστη επιλογή με βάση τα υπάρχοντα κελιά.

3.3.2.2 Καταχωρητές με ψηφοφόρο

Όπως αναφέρθηκε και παραπάνω, θα γίνουν δύο υλοποιήσεις του SPI slave. Μία που θα έχει τριπλό πλεονασμό μόνο στους καταχωρητές και μία που θα έχει τριπλό πλεονασμό σε όλα τα μέρη του. Και στις δύο περιπτώσεις θα αντικατασταθούν οι καταχωρητές με τις παρακάτω υλοποιήσεις.

Στην υλοποίηση με απλό TMR, κάθε καταχωρητής θα τριπλασιαστεί και στις εξόδους του θα προστεθεί ο ψηφοφόρος. Το σχηματικό του μετά την σύνθεση απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 3.8: Σχηματικό καταχωρητή με χρήση απλού TMR.

Στην υλοποίηση με πλήρες TMR, κάθε καταχωρητής θα τριπλασιαστεί και στις εξόδους τους θα προστεθούν τρείς ψηφοφόροι. Κάθε έξοδος του καταχωρητή, συνδέεται σε κάθε ψηφοφόρο, όπως φαίνεται στο σχήμα 3.9.

Κάθε καταχωρητής έχει σαν εισόδους το σήμα χρονισμού (clk), το σήμα επαναφοράς και την είσοδο δεδομένων. Σε κάποιες περιπτώσεις υπάρχει και σήμα ενεργοποίησης του καταχωρητή (enable). Σαν έξοδο έχει τον δίαυλο δεδομένων, που είναι η έξοδος του ψηφοφόρου. Ανά περίπτωση το μήκος του διαύλου δεδομένων εισόδου και εξόδου διαφέρει.



Σχήμα 3.9: Σχηματικό καταχωρητή με χρήση πλήρους TMR.

3.3.2.3 Μετρητής των bit διεύθυνσης

Υπάρχουν δύο μετρητές στον slave, οι οποίοι και οι δύο έχουν δύο καταχωρητές. Ένας χρειάζεται για να κρατάει τον αριθμό της αρίθμησης και η έξοδός του ανατροφοδοτείται στον εαυτό του μέσω ενός συνδυαστικού κυκλώματος που αυξάνει την τιμή του κατά ένα. Ο δεύτερος χρειάζεται για το σήμα ενεργοποίησης. Αυτός χρησιμοποιείται για να έχει καθυστέρηση το σήμα κατά ένα κύκλο ρολογιού, ώστε να υπάρχει σωστός συγχρονισμός του μετρητή με τις υπόλοιπες μονάδες. Αυτός ο καταχωρητής λειτουργεί στην ανοδική αιχμή του ρολογιού. Οι καταχωρητές χρησιμοποιούν TMR όπως αναφέρθηκε παραπάνω.

Αυτός ο μετρητής χρησιμοποιείται για να μετράει τους κύκλους ρολογιού που έρχονται τα 4 πρώτα bit που ορίζουν την διεύθυνση της μνήμης που θα γίνει ανάγνωση η εγγραφή και μετράει από το 0 έως το 3, δηλαδή χρειάζεται έξοδο με δίαυλο 2 bit. Έχει σαν εισόδους το σήμα χρονισμού (clk), το σήμα επαναφοράς (reset) και το σήμα ενεργοποίησης (enable), το οποίο όσο είναι '1' τότε ο μετρητής αυξάνεται κατά 1 σε κάθε κύκλο ρολογιού. Ο καταχωρητής για την αρίθμηση λειτουργεί κατά την καθοδική αιχμή του ρολογιού. Το σχηματικό μετά την σύνθεση με χρήση απλού TMR, φαίνεται στο παρακάτω σχήμα.



Σχήμα 3.10: Σχηματικό μετρητή κύκλων ρολογιού για την αποστολή διεύθυνσης με χρήση απλού TMR.

Βλέπουμε τους δύο καταχωρητές και το συνδυαστικό κύκλωμα που κάνει την πρόσθεση. Ο πίνακας αληθείας για το συνδυαστικό κύκλωμα είναι αυτός του πίνακα 2.

Πίνακας 2: Πίνακας αληθείας συνδυαστικού κυκλώματος του μετρητή των bit διεύθυνσης.

Είσα	οδοι	Έξο	δοι
Α	В	Y1	YO
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Από τον πίνακα αληθείας προκύπτουν οι λογικές συναρτήσεις για κάθε έξοδο. Για την Υ1 έχουμε:

$$Y\mathbf{1} = (\overline{A} \bullet B) + (A \bullet \overline{B}) = (A \oplus B)$$

Και για την Υ0 έχουμε:

$$Y\mathbf{0} = (\overline{A} \bullet \overline{B}) + (A \bullet \overline{B}) = \overline{B}$$

Αυτές τις δύο λογικές συναρτήσεις υλοποιεί το συνδυαστικό κύκλωμα του μετρητή. Στο σχηματικό της παραπάνω εικόνας, φαίνεται μόνο η πύλη XOR. Ο αντιστροφέας ενσωματώθηκε στο μπλοκ των καταχωρητών που αποθηκεύεται το αποτέλεσμα της αρίθμησης.

Στην υλοποίηση με πλήρες TMR, θα τριπλασιαστεί το συνδυαστικό κύκλωμα και επίσης οι είσοδοι και έξοδοι, όπως φαίνεται και στο σχήμα 3.11. Και σε αυτή την περίπτωση οι αντιστροφείς ενσωματώθηκαν μέσα στο μπλοκ καταχωρητών.



Σχήμα 3.11: Σχηματικό μετρητή κύκλων ρολογιού για την αποστολή διεύθυνσης με χρήση πλήρους TMR.

3.3.2.4 Μετρητής των bit δεδομένων

Αυτός ο μετρητής χρησιμοποιείται για την μέτρηση των κύκλων ρολογιού για την μεταφορά των 8 bit προς τον slave για την λειτουργία εγγραφής, ή προς τον master για τη λειτουργία ανάγνωσης. Όσο είναι ανενεργός έχει την τιμή μηδέν και ξεκινάει να μετράει από το ένα έως το οχτώ, δηλαδή χρειάζεται έξοδο με δίαυλο 4 bit. Έχει σαν εισόδους το σήμα χρονισμού (clk), το σήμα επαναφοράς (reset) και το σήμα ενεργοποίησης (enable), το οποίο όταν είναι '1' τότε ο μετρητής αυξάνεται κατά 1 σε κάθε κύκλο. Ο καταχωρητής για την αρίθμηση λειτουργεί κατά την ανοδική αιχμή του ρολογιού. Το σχηματικό μετά την σύνθεση με χρήση απλού TMR, φαίνεται στο παρακάτω σχήμα.



Σχήμα 3.12: Σχηματικό μετρητή κύκλων ρολογιού για την μεταφορά δεδομένων με χρήση απλού TMR.

Βλέπουμε τους δύο καταχωρητές και το συνδυαστικό κύκλωμα που κάνει την πρόσθεση. Ο πίνακας αληθείας για το συνδυαστικό κύκλωμα είναι αυτός του πίνακα 3.

	Είσα	οδοι			Έξο	δοι	
Α	В	С	D	Y3	Y2	Y1	YO
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

Πίνακας 3: Πίνακας αληθείας συνδυαστικού κυκλώματος του μετρητή των bit δεδομένων.

Από τον πίνακα αληθείας και απλοποίηση προκύπτουν οι λογικές συναρτήσεις για κάθε έξοδο. Για την Υ3 έχουμε:

$$Y3 = (A \bullet \overline{B}) + (A \bullet \overline{C}) + (A \bullet \overline{D}) + (\overline{A} \bullet B \bullet C \bullet D) = (A \bullet \overline{B} \bullet \overline{C} \bullet \overline{D}) + (\overline{A} \bullet B \bullet C \bullet D)$$
$$= A \bigoplus (C \bullet D \bullet B)$$

Για την Υ2 έχουμε:

$$Y2 = (\overline{B} \bullet \overline{C}) + (B \bullet \overline{D}) + (\overline{B} \bullet C \bullet D) = (C \bullet D) \oplus B$$

Για την Υ1 έχουμε:

$$Y\mathbf{1} = (\overline{C} \bullet D) + (C \bullet \overline{D}) = C \oplus D$$

Και τέλος για την Υ0 έχουμε:

 $Y0 = \overline{D}$

Όπως και στην περίπτωση του μετρητή των bit της διεύθυνσης, έτσι κι εδώ ο αντιστροφέας που υλοποιεί τη λογική συνάρτηση για το Y0 ενσωματώθηκε στο μπλοκ των κατάχωτών. Για την υλοποίηση των Y1, Y2 και Y3 το εργαλείο σύνθεσης χρησιμοποίησε δύο ημιαθροιστές και μία πύλη XOR, όπως δείχνει το σχήμα 3.12 παραπάνω.

Στην υλοποίηση με πλήρες TMR, θα τριπλασιαστεί το συνδυαστικό κύκλωμα και επίσης οι είσοδοι και έξοδοι, όπως φαίνεται και στο σχήμα 3.13. Και σε αυτή την περίπτωση οι αντιστροφείς ενσωματώθηκαν μέσα στο μπλοκ των καταχωρητών.



Σχήμα 3.13: Σχηματικό μετρητή κύκλων ρολογιού για την μεταφορά δεδομένων με χρήση πλήρους TMR.

3.3.2.5 Καταχωρητές ολίσθησης

Υπάρχουν δύο καταχωρητές ολίσθησης. Και οι δύο έχουν σαν εισόδους το ρολόι χρονισμού (clock), το σήμα ενεργοποίησης (enable), το σήμα επαναφοράς (reset) και την σειριακή είσοδο. Σαν έξοδο έχουν την παράλληλη έξοδο δεδομένων. Λειτουργούν στην

ανοδική αιχμή του ρολογιού για όσο το σήμα enable είναι '1'. Ο ένας καταχωρητής χρησιμοποιείται για να αποθηκεύει την διεύθυνση της μνήμης που θα προσπελαστεί. Η διεύθυνση αποτελείται από 4 bit, επομένως χρειάζονται 4 flip-flop, τα οποία συνδέονται το ένα μετά το άλλο, όπως δείχνει το σχήμα 3.14. Το κάθε bit έρχεται σειριακά, δηλαδή ένα σε κάθε κύκλο ρολογιού, με πρώτο το λιγότερο σημαντικό bit και σε κάθε κύκλο το κάθε bit σπρώχνεται μια θέση ώστε να συμπληρωθεί η διεύθυνση. Στην έξοδο του, τα δεδομένα στέλνονται παράλληλα στην θύρα o_data_out για να εγγραφούν στην μνήμη.



Σχήμα 3.14: Σχηματικό καταχωρητή ολίσθησης διευθύνσεων.

Στην υλοποίηση με απλό TMR, της οποίας το σχηματικό φαίνεται στο σχήμα 3.15, το κύκλωμα του καταχωρητή ολίσθησης του παραπάνω σχήματος, περιλαμβάνεται σε κάθε μπλοκ καταχωρητή.



Σχήμα 3.15: Σχηματικό καταχωρητή ολίσθησης με χρήση απλού TMR.

Στην υλοποίηση με πλήρες TMR, θα τριπλασιαστούν οι είσοδοι, οι έξοδοι και οι ψηφοφόροι, όπως φαίνεται και στο σχήμα 3.16. Όπως και στην υλοποίηση με απλό TMR, έτσι και στην υλοποίηση με πλήρες, το κύκλωμα του καταχωρητή ολίσθησης του σχήματος 3.14, περιλαμβάνεται σε κάθε μπλοκ καταχωρητή.



Σχήμα 3.16: Σχηματικό καταχωρητή ολίσθησης με χρήση πλήρους TMR.

Ο άλλος καταχωρητής ολίσθησης χρησιμοποιείται για να αποθηκεύει τα δεδομένα που θα γραφτούν στη μνήμη στη λειτουργία εγγραφής. Τα δεδομένα αποτελούνται από 8 bit, επομένως χρειάζονται 8 flip-flop, τα οποία συνδέονται το ένα μετά το άλλο, όπως δείχνει το σχήμα 3.17.



Σχήμα 3.17: Σχηματικό καταχωρητή ολίσθησης δεδομένων.

Κατά τα υπόλοιπα δεν διαφέρει σε τίποτα από τον καταχωρητή ολίσθησης διευθύνσεων και τα σχηματικά είναι τα ίδια, με την διαφορά ότι αντί για 4-bit οι δίαυλοι, είναι 8-bit.

3.3.2.6 Κεντρική μονάδα επεξεργασίας

Η κεντρική μονάδα επεξεργασίας είναι υπεύθυνη για τον σωστό συγχρονισμό όλων των υπομονάδων και την σωστή λειτουργία του slave. Η μονάδα επεξεργασίας αποτελείται από τρείς μικρές μηχανές πεπερασμένων καταστάσεων (Finite State Machines – FSM). Η μία FSM είναι υπεύθυνη για την ανάγνωση της διεύθυνσης μνήμης από τον master και έχει τέσσερις καταστάσεις. Η δεύτερη FSM είναι υπεύθυνη για την ανάγνωση δεδομένων από τη μνήμη και η τρίτη είναι υπεύθυνη για την εγγραφή δεδομένων στη μνήμη. Οι δύο

τελευταίες έχουν από δύο καταστάσεις. Επιλέχτηκε αυτή η υλοποίηση αντί για μία ενιαία FSM, λόγω του ότι οι τρείς μικρές είναι ταχύτερες λόγω του μικρότερου αριθμού συνδυαστικών κυκλωμάτων.

Η μονάδα ελέγχου έχει τις παρακάτω θύρες:

Είσοδοι

- MOSI: Σειριακή θύρα. Παίρνει τα δεδομένα από τον master.
- count_data: Δίαυλος 4-bit. Δέχεται την τιμή του μετρητή για τα δεδομένα.
- count_addr: Δίαυλος 2-bit. Δέχεται τα δεδομένα από τον μετρητή για τις διευθύνσεις.
- data_in: Δίαυλος 8-bit. Δέχεται τα δεδομένα από την μνήμη σε λειτουργία ανάγνωσης.
- CurrState: Δίαυλος 4-bit. Παίρνει την παρούσα κατάσταση από τον αντίστοιχο καταχωρητή καταστάσεων για τον FSM που είναι υπεύθυνος για την λήψη της διεύθυνσης.
- CurrState_wr: Δίαυλος 2-bit. Παίρνει την παρούσα κατάσταση από τον αντίστοιχο καταχωρητή καταστάσεων για τον FSM που είναι υπεύθυνος για τη λειτουργία εγγραφής.
- CurrState_rd: Δίαυλος 2-bit. Παίρνει την παρούσα κατάσταση από τον αντίστοιχο καταχωρητή καταστάσεων για τον FSM που είναι υπεύθυνος για τη λειτουργία ανάγνωσης.

Έξοδοι

- wr_en: Σήμα που επιτρέπει να γραφτεί δεδομένο στη μνήμη στη μνήμη.
- en_4bit: Σήμα ενεργοποίησης του καταχωρητή ολίσθησης διευθύνσεων. Όταν είναι '1', ενεργοποιείται ο καταχωρητής ολίσθησης.
- en_8bit: Σήμα ενεργοποίησης του καταχωρητή ολίσθησης δεδομένων. Όταν είναι
 '1', ενεργοποιείται ο καταχωρητής ολίσθησης.
- input_4bit: Στέλνει στον καταχωρητή ολίσθησης διευθύνσεων τα bit που αποτελούν τη διεύθυνση.
- input_8bit: Στέλνει στον καταχωρητή ολίσθησης δεδομένων τα bit των δεδομένων που θα εγγραφούν στη μνήμη.
- en_data_cnt: Σήμα ενεργοποίησης του μετρητή για τα bit των δεδομένων.
- en_addr_cnt: Σήμα ενεργοποίησης του μετρητή για τα bit της διεύθυνσης.
- **MISO:** Σειριακή θύρα. Στέλνει δεδομένα στον master.
- NextState Δίαυλος 4-bit. Στέλνει την επόμενη κατάσταση στον αντίστοιχο καταχωρητή καταστάσεων για τον FSM που είναι υπεύθυνος για την λήψη της διεύθυνσης.
- NextState_wr: Δίαυλος 2-bit. Στέλνει την επόμενη κατάσταση στον αντίστοιχο καταχωρητή καταστάσεων για τον FSM που είναι υπεύθυνος για τη λειτουργία εγγραφής.
- NextState_rd: Δίαυλος 2-bit. Στέλνει την επόμενη κατάσταση στον αντίστοιχο καταχωρητή καταστάσεων για τον FSM που είναι υπεύθυνος για τη λειτουργία ανάγνωσης.

Η πρώτη FSM είναι υπεύθυνη για την σωστή μεταφορά της διεύθυνσης στην έξοδο "o_addr_out" μέσω του καταχωρητή ολίσθησης διευθύνσεων. Το διάγραμμα καταστάσεων της, παρουσιάζεται στο σχήμα 3.18. Με το που γίνει το σήμα i_CSN του slave '0' και ενεργοποιηθεί, οι έξοδοι "en_4bit" και "en_addr_cnt" γίνονται '1' και ενεργοποιούνται ο μετρητής για τις διευθύνσεις και ο καταχωρητής διεύθυνσης διευθύνσεων. Επίσης στέλνει στον καταχωρητή ολίσθησης διευθύνσεων τα bit που έρχονται στην

είσοδο "MOSI", μέσω της εξόδου "input_4bit". Αυτά γίνονται από την κατάσταση "IDLE" επειδή το πρώτο bit του master είναι και το 1° bit της διεύθυνσης. Στον 2° κύκλο ρολογιού, η FSM μεταβαίνει στην κατάσταση "take_address" και παραμένει σε αυτή μέχρι ο μετρητής πάρει την τιμή '3' (4°ς κύκλος ρολογιού). Η κατάσταση των εξόδων δεν αλλάζει μέχρι τον 4° κύκλο ρολογιού, όπου η έξοδος "en addr cnt" γίνεται '0' και απενεργοποιείται ο μετρητής και η FSM μεταβαίνει στην επόμενη κατάσταση που είναι η "MODE". Σε αυτή την κατάσταση το σήμα "en data cnt" γίνεται '1' κι έτσι στον επόμενο κύκλο ρολογιού ξεκινάει να μετράει ο μετρητής για τα δεδομένα. Επίσης διαβάζει το bit (5° bit που δείχνει την λειτουργία) της εισόδου "MOSI" και αν είναι '0' τότε ακολουθεί λειτουργία εγγραφής και ενεργοποιεί την FSM που είναι υπεύθυνη για την εγγραφή, κάνοντας '1' το σήμα "move on wr". Σε αυτή την περίπτωση κάνει '1' και το σήμα "en_8bit", ώστε από τον επόμενο κύκλο ρολογιού να ενεργοποιηθεί ο καταχωρητής ολίσθησης για τα δεδομένα. Αν το MOSI είναι '1' τότε ακολουθεί λειτουργία ανάγνωσης και ενεργοποιεί την FSM που είναι υπεύθυνη για τη λειτουργία ανάγνωσης κάνοντας '1' το σήμα "move_on_rd". Σε αυτή την περίπτωση διαβάζει το περισσότερο σημαντικό ψηφίο από τον καταχωρητή, που έχει ήδη αποθηκεύσει το δεδομένο της θέσης μνήμης που αντιστοιχεί η διεύθυνση, μέσω της εισόδου "data in" και το στέλνει στην έξοδο MISO, ώστε στον επόμενο κύκλο ρολογιού να αποθηκευτεί στον αντίστοιχο καταχωρητή, μέσω του οποίου θα αποσταλεί το bit στον master από την σειριακή θύρα. Στον επόμενο κύκλο ρολογιού η FSM μεταβαίνει στην κατάσταση "waiting" και παραμένει σε αυτή τη κατάσταση έως ότου ολοκληρωθεί η εγγραφή ή η ανάγνωση, όπου θα γίνει επαναφορά και η FSM θα επιστρέψει στην αρχική της κατάσταση.



Σχήμα 3.18: Διάγραμμα καταστάσεων FSM για την αποθήκευση της διεύθυνσης.

Στον επόμενο πίνακα παρουσιάζεται ο πίνακας καταστάσεων της FSM που είναι υπεύθυνη για την αποθήκευση της διεύθυνσης. Σε περίπτωση απροσδιόριστης κατάστασης λόγω σφάλματος, τότε υπάρχει μετάβαση στην αρχική κατάσταση "IDLE". Για να μειωθεί η πιθανότητα μετάβασης σε λάθος κατάσταση σε περίπτωση σφάλματος, οι τέσσερις καταστάσεις κωδικοποιούνται ως εξής: "0000", "0011", "1100" και "1111".

Τρέχουσα κατάσταση	Συνθήκη		Επόμενη κατάσταση
IDLE	CO		TAKE_ADDRESS
TAKE_ADDRESS	C1	count_addr < 3	TAKE_ADDRESS
	C2	count_addr = 3	MODE
MODE	C3		WAITING
WAITING	C4		WAITING
OTHER CASE	C5		IDLE

Πίνακας 4: Πίνακας καταστάσεων FSM για την αποθήκευση διεύθυνσης.

Η δεύτερη FSM είναι υπεύθυνη για τη λειτουργία εγγραφής των δεδομένων στη μνήμη από την έξοδο "o_data_out" μέσω του καταχωρητή ολίσθησης δεδομένων. Το διάγραμμα καταστάσεων της παρουσιάζεται στο σχήμα 3.19. Όσο είναι '0' το σήμα "move_on_wr", η FSM βρίσκεται στην κατάσταση wr_IDLE. Με το που το σήμα πάρει την τιμή '1' μεταβαίνει στην κατάσταση "STORE_DATA" και θα παραμείνει σε αυτή μέχρι η είσοδος "count_data", που έρχεται από τον μετρητή για τα bit δεδομένων, πάρει την τιμή '8'. Σε όλη αυτή τη διάρκεια στέλνονται στον καταχωρητή ολίσθησης δεδομένων, μέσω της θύρας "input_8bit", τα bit των δεδομένων που λαμβάνονται από τον μετρητής φτάσει στο 8, τότε η έξοδος "wr_en" γίνεται για έναν κύκλο ρολογιού '1' και γράφονται στη μνήμη τα δεδομένα. Στον επόμενο κύκλο ρολογιού η FSM μεταβαίνει στην κατάσταση wr_IDLE. Σε περίπτωση απροσδιόριστης κατάστασης λόγω σφάλματος, τότε υπάρχει μετάβαση στην αρχική κατάσταση "IDLE". Για να μειωθεί η πιθανότητα μετάβασης σε λάθος κατάσταση σε περίπτωση σφάλματος, οι δύο καταστάσεις κωδικοποιούνται ως εξής: "01" και "10".



Σχήμα 3.19: Διάγραμμα καταστάσεων FSM για την εγγραφή δεδομένων.

Στον επόμενο πίνακα παρουσιάζεται ο πίνακας καταστάσεων της FSM που είναι υπεύθυνη για την εγγραφή των δεδομένων στη μνήμη. Σε περίπτωση απροσδιόριστης κατάστασης λόγω σφάλματος, τότε υπάρχει μετάβαση στην αρχική κατάσταση "wr_IDLE".

Τρέχουσα κατάσταση	Συνθήκη		Επόμενη κατάσταση
wr_IDLE -	CO	move_on_wr = '0'	wr_IDLE
	C1	move_on_wr = '1'	STORE_DATA
STORE_DATA	C2	count_data < ή = 8	STORE_DATA
	C3	count_data > 8	wr_IDLE
OTHER_STATES	C4		wr_IDLE

Πίνακας 5: Πίνακας καταστάσεων FSM για την εγγραφή δεδομένων στη μνήμη.

Η τρίτη FSM είναι υπεύθυνη για τη λειτουργία ανάγνωσης των δεδομένων της μνήμης από τη θύρα "data_in". Το διάγραμμα καταστάσεων της παρουσιάζεται στο σχήμα 3.20. Όσο είναι '0' το σήμα "move_on_rd", η FSM βρίσκεται στην κατάσταση rd_IDLE. Με το που το σήμα πάρει την τιμή '1' μεταβαίνει στην κατάσταση "READ_DATA" και θα παραμείνει σε αυτή μέχρι η είσοδος "count_data", που έρχεται από τον μετρητή για τα bit δεδομένων, πάρει την τιμή '7'. Σε όλη αυτή τη διάρκεια, σε κάθε κύκλο ρολογιού διαβάζεται κάθε bit που έρχεται από τον καταχωρητή εισόδου και στέλνεται στον καταχωρητή "MISO", μέσω της αντίστοιχης θύρας. Όταν ο μετρητής φτάσει στο 7, στον επόμενο κύκλο ρολογιού η FSM μεταβαίνει στην κατάσταση wr_IDLE. Σε περίπτωση απροσδιόριστης κατάστασης λόγω σφάλματος, τότε υπάρχει μετάβαση στην αρχική κατάσταση "IDLE". Για να μειωθεί η πιθανότητα μετάβασης σε λάθος κατάσταση σε περίπτωση σφάλματος, οι δύο καταστάσεις κωδικοποιούνται ως εξής: "01" και "10".



Σχήμα 3.20: Διάγραμμα καταστάσεων FSM για την ανάγνωση δεδομένων.

Στον επόμενο πίνακα παρουσιάζεται ο πίνακας καταστάσεων της FSM που είναι υπεύθυνη για την ανάγνωση των δεδομένων. Σε περίπτωση απροσδιόριστης κατάστασης λόγω σφάλματος, τότε υπάρχει μετάβαση στην αρχική κατάσταση "rd_IDLE".

The area of the ar
--

Επόμενη κατάσταση		Συνθήκη	Παρούσα κατάσταση
rd_IDLE	C0	move_on_rd = '0'	rd_IDLE
	C1	move_on_rd = '1'	READ_DATA
READ_DATA	C2	count_data < ή = 7	READ_DATA
	C3	count_data > 7	rd_IDLE
Other states	C4		rd_IDLE

Το σχηματικό της μονάδας ελέγχου μετά την σύνθεση, απεικονίζεται στο σχήμα 3.21.



Σχήμα 3.21: Σχηματικό κεντρικής μονάδας επεξεργασίας του slave.

3.3.2.7 Ανώτερο επίπεδο του slave (Top level)

Στο ανώτερο επίπεδο συνδέονται όλες οι υποομάδες που αναπτύξαμε παραπάνω και αποτελεί την συνολική σύνθεση του slave. Η υλοποίηση με χρήση απλού TMR περιλαμβάνει τις θύρες που αναφέραμε στην παράγραφο 3.3.1 και το σχηματικό της απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 3.22: Σχηματικό SPI slave με χρήση απλού TMR.

Εκτός από τους μετρητές, τους καταχωρητές ολίσθησης και την μονάδα ελέγχου, ο slave περιέχει και καταχωρητές για τα σήματα εξόδου wr_en και MISO, για την είσοδο data_in.

Επίσης περιλαμβάνει τους καταχωρητές καταστάσεων, έναν για κάθε FSM. Τέλος έχει και μία πύλη NOR κι έναν αντιστροφέα που υλοποιούν το σήμα επαναφοράς. O slave είναι ενεργός μόνο όταν το "reset" είναι '1' και το "i_CSN" είναι '0'. Σε οποιαδήποτε άλλη περίπτωση ενεργοποιείται το σήμα επαναφοράς. Όπως είπαμε και παραπάνω, στην υλοποίηση με χρήση απλού TMR, πλεονασμός υπάρχει μόνο στους καταχωρητές.

Η υλοποίηση με πλήρες TMR εκτός από τους καταχωρητές, έχει πλεονασμό και στις θύρες και στο συνδυαστικό κομμάτι και στους ψηφοφόρους. Συγκεκριμένα για το συνδυαστικό κομμάτι, υπάρχουν τρείς κεντρικές μονάδες επεξεργασίας και τρείς ψηφοφόροι για κάθε σήμα ενεργοποίησης των μετρητών και των καταχωρητών ολίσθησης. Οι υπόλοιπες έξοδοί τους καταλήγουν σε κάποιον καταχωρητή και κατ' επέκταση σε ψηφοφόρο. Το σχηματικό του slave με χρήση πλήρους TMR απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 3.23: Σχηματικό SPI slave με χρήση πλήρους TMR.

3.3.3 Φυσική σχεδίαση (layout) του slave

Μετά τη σύνθεση του slave και την εξαγωγή των κατάλληλων αρχείων, ακολουθεί η φυσική σχεδίαση, η οποία έγινε με το εργαλείο Innovus της Cadence, στην έκδοση v15.20.

Η φυσική σχεδίαση χωρίζεται σε γραμμές, όπου σε κάθε γραμμή τοποθετούνται τα κελιά. Οι τροφοδοσίες των κελιών κάθε γραμμής συνδέονται μεταξύ τους και η τροφοδοσία της κάθε γραμμής συνδέεται στο στεφάνι που περιβάλλει τα κελιά.

3.3.3.1 Φυσική σχεδίαση του slave με χρήση απλού TMR

Στο σχήμα 3.24 απεικονίζεται η φυσική σχεδίαση του slave σε υλοποίηση με χρήση απλού TMR. Χρησιμοποιήθηκαν τρία επίπεδα μετάλλων.


Σχήμα 3.24: Φυσική σχεδίαση του slave σε υλοποίηση με χρήση απλού TMR.



Σχήμα 3.25: Δέντρο ρολογιού του slave για υλοποίηση με χρήση απλού TMR.

Το πρώτο επίπεδο (μπλε χρώμα) χρησιμοποιείται για τις συνδέσεις τις τροφοδοσίας. Το δεύτερο επίπεδο (κόκκινο χρώμα) χρησιμοποιείται για τις συνδέσεις εντός των κελιών, αλλά και μεταξύ των και τέλος το τρίτο επίπεδο (πράσινο χρώμα) χρησιμοποιείται για συνδέσεις μεταξύ των κελιών.

Στο σχήμα 3.25 απεικονίζεται η σύνθεση του δέντρου ρολογιού, δηλαδή την κατανομή του σήματος χρονισμού μέσα στο μπλοκ του slave. Το σήμα χρονισμού βλέπουμε ότι έχει χωριστεί σε οχτώ τομείς. Με κίτρινο χρώμα βλέπουμε το σήμα χρονισμού από τον ακροδέκτη "i_SCLK" να μοιράζεται σε οχτώ buffers (κελί CLKBUFF_X1), όπου μέσω του κάθε buffer στέλνεται το σήμα χρονισμού σε κάθε flip-flop του τομέα. Ο κάθε τομέας απεικονίζεται με διαφορετικό χρώμα.

Στο σχήμα 3.26 βλέπουμε πως εχουν κατανεμηθεί τα κελιά και οι μονάδες στο μπλοκ του slave. Παρατηρούμε ότι ο καταχωρητής ολίσθησης δεδομένων (πράσινο σκούρο χρώμα) και ο καταχωρητής εισόδου δεδομένων από τη μνήμη (μπλε χρώμα) καταλαμβάνουν τον περισσότερο χώρο. Αυτό είναι λογικό αφού είναι οι μεγαλύτεροι καταχωρητές (8-bit), επομένως χρειάζονται 8 flip-flop και λόγω του ότι χρησιμοποιείται πλεονασμός, αυτά τριπλασιάζονται. Δηλαδή 8 x 3 = 24 flip-flops συν τα κελιά που χρησιμοποιούνται για να υλοποιήσουν τους ψηφοφόρους.



Σχήμα 3.26: Κατανομή κελιών και μονάδων του slave για υλοποίηση με χρήση απλού TMR.

3.3.3.2 Φυσική σχεδίαση (layout) του slave με χρήση πλήρους TMR

Στο σχήμα 3.27 απεικονίζεται η φυσική σχεδίαση του slave σε υλοποίηση με χρήση πλήρους TMR. Σε αντίθεση με την υλοποίηση με απλό TMR, χρησιμοποιήθηκαν τέσσερα επίπεδα μετάλλων. Το πρώτο επίπεδο (μπλε χρώμα) χρησιμοποιείται για τις συνδέσεις τις τροφοδοσίας. Το δεύτερο επίπεδο (κόκκινο χρώμα) χρησιμοποιείται για τις συνδέσεις εντός των κελιών, αλλά και μεταξύ των και τέλος το τρίτο επίπεδο (πράσινο χρώμα) και τέταρτο επίπεδο (κίτρινο χρώμα) χρησιμοποιούνται για συνδέσεις μεταξύ των κελιών.



Σχήμα 3.27: Φυσική σχεδίαση του slave σε υλοποίηση με χρήση πλήρους TMR.



Σχήμα 3.28: Δέντρο ρολογιού του slave για υλοποίηση με χρήση πλήρους TMR.

Στο σχήμα 3.28 απεικονίζεται η σύνθεση του δέντρου των ρολογιών. Εδώ σε αντίθεση με την υλοποίηση απλού TMR, έχουμε τρία σήματα χρονισμού. Με κίτρινο χρώμα βλέπουμε το "i_SCLK1", το οποίο χωρίζεται σε τέσσερις τομείς, με κόκκινο βλέπουμε το i_SCLK2", το οποίο χωρίζεται και αυτό σε τέσσερις τομείς και τέλος με πράσινο το "i_SCLK3", το οποίο χωρίζεται σε τρείς τομείς.

Στο σχήμα 3.29 βλέπουμε πως εχουν κατανεμηθεί τα κελιά και οι μονάδες στο μπλοκ του slave. Παρατηρούμε ότι ο καταχωρητής ολίσθησης δεδομένων (πράσινο σκούρο χρώμα) και ο καταχωρητής εισόδου δεδομένων από τη μνήμη (κόκκινο χρώμα) καταλαμβάνουν τον περισσότερο χώρο όπως και στην υλοποίηση με χρήση απλού TMR. Εδώ αυξήθηκαν οι επιφάνειες τους λόγω πλεονασμού και στους ψηφοφόρους. Παρόλο που τώρα έχουμε πλεονασμό και στο συνδυαστικό κομμάτι της Κ.Μ.Ε (κίτρινο χρώμα) και είναι περισσότερα σε αριθμό τα κελιά που χρησιμοποιούνται δεν ξεπερνάει σε επιφάνεια τους δύο καταχωρητές. Αυτό συμβαίνει λόγω του ότι δεν έχει καθόλου flip-flops, τα οποία καταλαμβάνουν αρκετά μεγαλύτερη επιφάνεια από τις πύλες.



Σχήμα 3.29: Κατανομή κελιών και μονάδων του slave για υλοποίηση με χρήση πλήρους TMR.

3.4 Προσομοιώσεις

Στην ενότητα αυτή θα δούμε τις κυματομορφές από τις προσομοιώσεις του slave. Για τις προσομοιώσεις θα χρησιμοποιηθεί το εργαλείο της Cadence Simvision με έκδοση v15.10. Λόγω του ότι τα σήματα είναι τα ίδια και στις δύο υλοποιήσεις, θα δούμε τις κυματομορφές μόνο της υλοποίησης με πλήρες TMR.

Αρχικά, στο σχήμα 3.30, θα δούμε τις κυματομορφές των καταχωρητών με ψηφοφόρους. Με πράσινο χρώμα είναι το σήμα χρονισμού, το σήμα επαναφοράς και το σήμα ενεργοποίησης. Με κίτρινο είναι οι είσοδοι των τριών καταχωρητών και με γαλάζιο οι έξοδοι των τριών ψηφοφόρων. Βλέπουμε ότι σε κάθε σετ δεδομένων στην έξοδο παίρνουμε πάντα την είσοδο που υπάρχει σε πλειοψηφία.

🔍 Baseline 🕶 = 0 💾 Cursor-Baseline 🕶 = 0		Baseline = 0 TimeA = 0									
Name 🗢 🗸	Curso 🗢 🔻	0 20ns	. 40	ns .	60	Ins	80	Ins	10	Ons	120
	0										
······• 🚾 · rst	0										
- en	0										
🕂 🛲 Register1_in[7:0]	'h 00	00	10	90	В1	A 3	DD	4A	65	55	F8
	'h 00	00	10	90	B1	33	DD	44	78	50	53
⊕ ≣	'h 00	00	14	99	вв	A3	33	4A	78	50	в7
	'h 00	00		90	В1	A 3	DD	4A	78	50	F3
	'h 00	00		90	В1	A3	DD	4A	78	50	F3
	'h 00	00		90	В1	A3	DD	4A	78	50	F3

Σχήμα 3.30: Κυματομορφές καταχωρητών με ψηφοφόρους.

Στην τελευταία περίπτωση που έχουμε τρείς διαφορετικές εισόδους στους καταχωρητές, παίρνουμε ένα αποτέλεσμα που δεν ταιριάζει στην είσοδο κανενός καταχωρητή. Αυτό συμβαίνει για το λόγο πως η ψηφοφορία γίνεται σε κάθε bit του καταχωρητή. Το "F3" που προκύπτει εξηγείται στον πίνακα 7 που ακολουθεί.

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Hex
Register 1	1	1	1	1	1	0	0	0	F8
Register 2	0	1	0	1	0	0	1	1	53
Register 3	1	0	1	1	0	1	1	1	B7
Voter out	1	1	1	1	0	0	1	1	F3

Πίνακας 7: Λειτουργία ψηφοφόρου.

Για την προσομοίωση του συνολικού slave, χρειάστηκε να φτιάξουμε σε επίπεδο προσομοίωσης την λειτουργία του master. Στο σχήμα 3.31 βλέπουμε τις κυματομορφές για την λειτουργία εγγραφής στη μνήμη. Ο master στέλνει τη λέξη "01010101" ("55" στο δεκαεξαδικό) να γραφτεί στη θέση μνήμης "1011" ("Β" στο δεκαεξαδικό).

Με πράσινο χρώμα είναι το σήμα χρονισμού που έρχεται από τον master, το σήμα επαναφοράς (reset) και το σήμα ενεργοποίησης του slave (i_CSN). Όταν γίνεται '0' το σήμα "i_CSN" (κόκκινος δείκτης) ξεκινάει η επικοινωνία μεταξύ master και slave. Τα bit της διεύθυνσης έρχονται ένα – ένα μέσω της θύρας MOSI. Λόγω πλήρους υλοποίησης TMR έχουμε τριπλασιασμό και στις θύρες. Οι τρείς MOSI απεικονίζονται με κόκκινο χρώμα. Εσκεμμένα στην θύρα "MOSI3" βάλαμε διαφορετικά δεδομένα για να ελέγξουμε τη αποτελεσματικότητα του TMR. Παρόλο που έρχονται λανθασμένα δεδομένα από μία είσοδο, βλέπουμε (με κίτρινο χρώμα) ότι στους τρείς καταχωρητές ολίσθησης διορθώνεται το σφάλμα και αποθηκεύουν όλοι τη σωστή πληροφορία.

Το πέμπτο bit που έρχεται (μπλε δείκτης) καθορίζει τη λειτουργία, δηλαδή εγγραφή ή ανάγνωση. Η είσοδος "MOSI3" είναι '1', που σημαίνει ότι καθορίζει λειτουργία ανάγνωσης. Το σφάλμα που προκαλείται στον τρίτο FSM, διορθώνεται έμμεσα μέσω των ψηφοφόρων που υπάρχει στους καταχωρητές κατάστασης και ανατροφοδοτείται στον FSM. Αυτό φαίνεται αφού οι καταχωρητές κατάστασης για την εγγραφή και των τριών FSM (μπλε χρώμα) έχουν την ίδια τιμή.

Τέλος τα τελευταία 8 bit είναι τα δεδομένα που θα εγγραφούν στη μνήμη. Η θύρα "MOSI3" στέλνει λανθασμένα δεδομένα, αλλά πάλι βλέπουμε (γαλάζιο χρώμα) ότι οι καταχωρητές

ολίσθησης των δεδομένων αποθηκεύουν τη σωστή πληροφορία. Όταν ο μετρητής των bit για τα δεδομένα πάρει την τιμή '8', η έξοδος write_enable γίνεται '1' και βλέπουμε (με άσπρο χρώμα) ότι στην θέση "1011" (11 στο δεκαδικό) αποθηκεύεται το "01010101" ("55" στο δεκαεξαδικό).



Σχήμα 3.31: Κυματομορφές προσομοίωσης για την λειτουργία εγγραφής.

Στο σχήμα 3.32 βλέπουμε τις κυματομορφές για την λειτουργία ανάγνωσης από τη μνήμη. Ο master ζητάει τη λέξη "01010101" ("55" στο δεκαεξαδικό) που βρίσκεται στη θέση μνήμης "1011" ("Β" στο δεκαεξαδικό).

Με πράσινο χρώμα είναι το σήμα χρονισμού που έρχεται από τον master, το σήμα επαναφοράς (reset) και το σήμα ενεργοποίησης του slave (i_CSN). Όταν γίνεται '0' το σήμα "i_CSN" (κόκκινος δείκτης) ξεκινάει η επικοινωνία μεταξύ master και slave. Τα bit της διεύθυνσης έρχονται ένα – ένα μέσω της θύρας MOSI. Λόγω πλήρους υλοποίησης TMR έχουμε τριπλασιασμό και στις θύρες. Οι τρείς MOSI απεικονίζονται με κόκκινο χρώμα. Εσκεμμένα στην θύρα "MOSI3" βάλαμε διαφορετικά δεδομένα για να ελέγξουμε τη αποτελεσματικότητα του TMR. Παρόλο που έρχονται λανθασμένα δεδομένα από μία είσοδο, βλέπουμε (με κίτρινο χρώμα) ότι στους τρείς καταχωρητές ολίσθησης διορθώνεται το σφάλμα και αποθηκεύουν όλοι τη σωστή πληροφορία.

Το πέμπτο bit που έρχεται (μπλε δείκτης) καθορίζει τη λειτουργία, δηλαδή εγγραφή ή ανάγνωση. Η είσοδοι "MOSI" είναι ΄0', που σημαίνει ότι έχουμε λειτουργία ανάγνωσης. Ο καταχωρητής εισόδου (ροζ χρώμα) αποθηκεύει τα δεδομένα της μνήμης για την συγκεκριμένη διεύθυνση.

Τέλος, στον έκτο κύκλο ρολογιού (πορτοκαλί δείκτης) ξεκινάει η ανάγνωση των δεδομένων της μνήμης. Για τους επόμενους οχτώ κύκλους ρολογιού, μεταφέρονται ένα – ένα τα bit από τον καταχωρητή που τα έχει αποθηκευμένα προς την έξοδο "o_MISO", μέσω του καταχωρητή "MISO_reg". Τα δεδομένα στέλνονται ξεκινώντας από το υψυλότερης σημασίας bit.

Baseline ▼= 0			Marke	er 1 = 1300ns 310ns		
r Cursor-Baseline ▼= 1260ns		Time A = 1280 mc				
Name 🗢	Curs o 🗸	1260ns 1280ns	1300r	ns 1320ns	1340ns	1360ns 1380ns
	1					
⊕ ⊡ CSN	0					
	0					
	1					
	1					
	1					
	'Ъ 0000	0000 0011	1100	1111		
	'h 0	0 1 2	3)		
	'h 0	0 1 2 5	В			
	'h 0	0 1 2 5	В			
	'h 0	0 1 2 5	В			
🕀 🛲 Write function						
Read_function						
⊕… 📰 Read_state_reg1[1:0]	'h 0	0 (1		2		
⊕… · Read_state_reg2[1:0]	'h 0	0 1		2		
⊕… ≣ Read_state_reg3[1:0]	'h 0	0 (1		2		
⊕… 🛲 Data_in_Reg[7:0]	'h 00	00 0 F	00 55			
⊕ 📰 Data_counter [3:0]	'h 0	0		1 2 3	4 (5	6 (7 8 (9
	0					
	[16 x ▶	[16 x 8 bits]				
Hemory[15]	'h C9	C9				
	'h 00	00				
	'h 00	00				
⊕	'h 00	00				
⊕¶an Memory[11]	'h 55	55				

Σχήμα 3.32: Κυματομορφές προσομοίωσης για την λειτουργία ανάγνωσης.

3.5 Αποτελέσματα σύνθεσης και φυσικής σχεδίασης

Σε αυτή την ενότητα θα παρουσιαστούν τα αποτελέσματα της σύνθεσης και της φυσικής σχεδίασης των δύο υλοποιήσεων για τα ακρότατα σημεία TT, SS και FF. Για να έχουμε ένα μέτρο σύγκρισης για τα τροποποιημένα κελιά, θα κάνουμε τις δύο υλοποιήσεις του slave και με τα πρωτότυπα κελιά, δηλαδή χωρίς δακτυλίους προστασίας και θα συγκρίνουμε τα αποτελέσματα. Θα συγκρίνουμε μόνο για το σημείο TT.

Για όλες τις υλοποιήσεις χρησιμοποιήσαμε σήμα χρονισμού (ρολόι) με περίοδο 12 ns λόγω του ότι είναι ο ελάχιστος χρόνος που χρειάζεται η πιο αργή περίπτωση που είναι η υλοποίηση με χρήση πλήρους TMR με τροποποιημένα κελιά για ακρότατο σημείο SS.

3.5.1 Αποτελέσματα για την υλοποίηση με χρήση απλού TMR

Για την υλοποίηση με απλό TMR συνολικά χρησιμοποιήθηκαν 371 κελιά με συνολική επιφάνεια 19.393 μm² για την υλοποίηση με τα τροποποιημένα κελιά, ενώ για την υλοποίηση με τα πρωτότυπα κελιά χρησιμοποιήθηκαν 370 κελιά με συνολική επιφάνεια 9726 μm². Στο σχήμα 3.33 (α) βλέπουμε τα στατιστικά για την υλοποίηση με δακτυλίους προστασίας, ενώ στο σχήμα 3.33 (β) για την υλοποίηση με τα πρωτότυπα κελιά. Στον αριθμό των buffer δεν περιλαμβάνονται οι 8 buffers που χρησιμοποιήθηκαν για το δέντρο του ρολογιού. Παρατηρούμε μικρές αλλαγές ως προς τον αριθμό κάποιων κελιών που χρησιμοποιήθηκαν. Αυτό οφείλεται στο ότι η βιβλιοθήκη έχει διαφορετικούς χρόνους καθυστέρησης, κατανάλωσης ισχύος κλπ. Έτσι το εργαλείο σύνθεσης επέλεξε τη βέλτιστη περίπτωση.

Επίσης βλέπουμε ότι η συνολική επιφάνεια είναι σχεδόν διπλάσια στην υλοποίηση με τα τροποποιημένα κελιά. Αυτό είναι λογικό διότι το κάθε τροποποιημένο κελί είναι διπλάσιο από το αντίστοιχο πρωτότυπο.

ακτινοβ	ολίες
---------	-------

Gate	Instances	Area	Library	Gate	Instances	Area	Library
*********	*********	***********		*********		*********	
AND2_X0d5	24	578.534	<pre>Iv8hvt_guardring_tt_1.8V_25</pre>	ANO2_X0d5	35	421.848	lf15adhvt9s_typical
A021_X0d5	43	1295.676	<pre>Iv8hvt_guardring_tt_1.8V_25</pre>	A021 X8d5	44	662.984	lf15adhvt9s_typical
A0121_X0d5	8	192.845	1v8hvt guardring tt 1.8V 25	A0121_X0d5	7	84.370	lf15adhvt9s_typical
CLKBUF_X1	14	253.109	1v8hvt guardring tt 1.8V 25	CLKBUF X1	14	126.554	lf15adhvt9s typical
DFCS X1	36	5423.760	1v8hvt guardring tt 1.8V 25	DFCS X1	36	2711.8B0	lf15adhvt9s typical
DFC X1	21	2657.642	1v8hvt guardring tt 1.8V 25	DFC X1	21	1328.821	lf15adhvt9s typical
DFNC X1	57	6183.086	1v8hvt guardring tt 1.8V 25	DFNC X1	57	3091.543	lf15adhvt9s typical
HA XI	2	120.528	1v8hvt guardring tt 1.8V 25	HA XI	2	68.264	lf15adhvt9s typical
INV X1	66	795.485	1v8hvt guardring tt 1.8V 25	INV X1	65	391.716	lf15adhvt9s typical
NOR2 X8d5	96	1735.603	1v8hvt guardring tt 1.8V 25	NOR2 X8d5	85	768.366	lf15adhvt9s typical
0AI22 X8d5	2	68.264	1v8hvt guardring tt 1.8V 25	0A122 X8d5	2	30.132	lf15adhvt9s typical
X0R2_X1	2	96.422	1v8hvt_guardring_tt_1.8V_25	XOR2_X1	2	48.211	lf15adhvt9s_typical
				*********		*********	
total	371	19392.955		total	370	9726.610	
Туре	Instances	Area Are	a %	Type	Instances	Area Are	ta %
*********	*********	*********					
sequential	114 1	4264.489 7	3.6	sequential	114 7	132.244	73.3
inverter	66	795.485	4.1	inverter	65	391.716	4.0
buffer	14	253.109	1.3	buffer	14	126.554	1.3
logic	177	4079.873 2	1.0	logic	177 2	076.095	21.3
*********	*********	**********	****				
total	371 1	9392.955 18	0.0	total	370 5	726.610 1	90.0
		(α)				(B)	

Σχήμα 3.33: Στατιστικά για τα κελιά που χρησιμοποιήθηκαν και για την συνολική επιφάνεια σε απλό TMR για υλοποίηση: (α) με τα τροποποιημένα κελιά. (β) με τα πρωτότυπα κελιά.

Στην εικόνα 3.34 βλέπουμε το κρίσιμο μονοπάτι, δηλαδή το πιο αργό κομμάτι του κυκλώματος, (α) για υλοποίηση με τα τροποποιημένα κελιά και (β) για υλοποίηση με τα πρωτότυπα κελιά.

Για την υλοποίηση με τα τροποποιημένα κελιά το κρίσιμο μονοπάτι είναι 3,506 ns. Ξεκινάει από το ρολόι του καταχωρητή κατάστασης για την αποθήκευση της διεύθυνσης και καταλήγει στην είσοδο του σήματος ενεργοποίησης του καταχωρητή ολίσθησης δεδομένων. Ο καταχωρητής κατάστασης λειτουργεί στην καθοδική αιχμή του ρολογιού, οπότε το κρίσιμο μονοπάτι ξεκινάει από τα 6 ns και ολοκληρώνεται στα 9,506 ns και απομένουν 2,494 ns μέχρι την ολοκλήρωση της περιόδου του παλμού του ρολογιού και την έλευση της θετικής ανόδου που λειτουργεί ο καταχωρητής ολίσθησης δεδομένων.

Για την υλοποίηση με τα πρωτότυπα κελιά το κρίσιμο μονοπάτι είναι 3,017 ns. Ξεκινάει από το ρολόι του καταχωρητή κατάστασης για την αποθήκευση της διεύθυνσης και καταλήγει στην έξοδο του καταχωρητή του μετρητή για τα bit διεύθυνσης. Ο καταχωρητής κατάστασης λειτουργεί στην καθοδική αιχμή του ρολογιού, οπότε το κρίσιμο μονοπάτι ξεκινάει από τα 6 ns και ολοκληρώνεται στα 9,017 ns και απομένουν 2,983 ns μέχρι την ολοκλήρωση της περιόδου του παλμού του ρολογιού και την έλευση της θετικής ανόδου που λειτουργεί ο μετρητής των bit της διεύθυνσης.

Στην εικόνα 3.35 βλέπουμε το κρίσιμο μονοπάτι για υλοποίηση με τα τροποποιημένα κελιά, (α) για ακρότατο σημείο FF στους -40° C και (β) για ακρότατο σημείο SS στους 70° C.

Και στις δύο περιπτώσεις το κρίσιμο μονοπάτι είναι το ίδιο. Ξεκινάει από το ρολόι του καταχωρητή κατάστασης για την αποθήκευση της διεύθυνσης και καταλήγει στην είσοδο του σήματος ενεργοποίησης του καταχωρητή ολίσθησης δεδομένων. Για ακρότατο σημείο FF το κρίσιμο μονοπάτι είναι 2,666 ns ενώ για SS είναι 5,626 ns. Ο καταχωρητής κατάστασης λειτουργεί στην καθοδική αιχμή του ρολογιού, οπότε το κρίσιμο μονοπάτι ξεκινάει από τα 6 ns και ολοκληρώνεται στα 8,666 ns για FF και 11,626 ns για SS. Απομένουν 3,334 ns για FF και 0,374 ns για SS μέχρι την ολοκλήρωση της περιόδου του παλμού του ρολογιού και την έλευση της θετικής ανόδου που λειτουργεί ο καταχωρητής ολίσθησης δεδομένων.

	Cost Group : 'cik' (path_group 'cik') Timing slack : 2494ps Start-point : take_addr_state_reg/TMMR_Si End-point : data_shift_reg/Shift_regs[(clock clk) capture	<pre>date_shift_reg/en3 shift_reg[2].Shift_register/en Shift_reg[8].Shift_register/en Dout_reg/SE <<< DFC5_X1 Dout_reg/CP setup</pre>	g1468/0 INV_X1 2 FSM/en_Bbit	9141L/01 A0121_X8d5	g1425/0 AND2_x9d5	g1427/0 NOR2_X0d5 01425/A1	g1448/0 INV_X1 g1427/A1	g1455/Q NOR2_X0d5	<pre>take_addr_state_reg/Data_out1[0] FSM/CurrState[0] g1455/A2</pre>	g130/0 A021_X0d5 TMR Simple.voter unit/voter out[0]	g134/0 NOR2_X8d5 g138/81	g138/Q NOR2_X8d5 g134/A1	<pre>TMR_Simple.Regs[0].Reg_units/Dout[0] TMR_Simple.voter_unit/voter_ini[0] gl38/A2</pre>	Dout_reg[0]/CP DFNC_X1	take addr_state_reg TMR Simple Regial Reg units	Pin Type Fanou	Module: spl_slave_TMR Technology Library: ivBhvt_guardrin Operating conditions: PVT_IP8V_25C Interconnect mode: global Area mode: physical librar
(9)	mple.Re 2].Shii		4 9- 8	4 243.7	1 15.8	2 23.5	1 12.8	3 29.5	2 23.0		2 20.5	1 12.9			2 20.0		t Load (fF)	g_tt()
	egs[0]. ft_regi			7 1513	8 254	338	409	5 232	5 678		180	9 169	438		5 90		d Slew (ps)	.8V_25
	Reg_un ster/s		- 375 - 5	+815	+164	+298	+264	+ 185	481	÷	+376	+147	+258	÷	+227		(ps)	1.0
	its/Dout_ hift_reg[12000 R	9138 9586 R	9130 R	8315 F	8150 R	7860 R	7596 F	7411 R	7010	7009 F	6633 F	6486 R	6227	6000 F	1 0000	Arrival (ps)	
	Cost Group Timing slack ter/Dout_reg/SE Start-point End-point	(clock clk)	esMyen_addr_g cnt_en_reg Dout_re Dout_re	g1484/81 g1484/0	g1412/A g1412/A	g1414/AL	g1429/A g1429/0	g1434/A1 g1434/0	g1451/A2 g1451/0	TMR_Simple FSM/CurrState	0/0E150 130/01	g134/A1 g134/Q	g138/0 g138/0	Dout_reg TMR_Simple TMR_Simple	TMR_Simple Dout_reg	(clock clk) take addr sta	Pin	Technology Operating o Interconnec Area mode:
	: 'clk' : 290 : take_a : addres		nt pata_in3 Data_in3 e.Regs[2 e.Regs[2 g/CP							voter_un ite_reg/0				Regs[8]. voter_un	Regs [0]. 0]/CP	ite reg	Ē	librarie ondition t mode:
	<pre>(path_group 'cl igps s_counter/cnt_e</pre>	capture	 }.Reg_units/Dir << DFC_X1 setup	A0121_X8d5	TIM X1	AND AND	INV X1	NOR2_X9d5	NOR2 X8d5	htt/voter_out[0]	A021_X0d5	NOR2_X8d5	NOR2_X0d5	DFNC_X1 Reg_units/Dout[it/voter_in1[0]	Reg_units	launch	Type	spi_slave physical_c s: typical global physical 1
(B	INR_SIN				- t	;		N	N		N		н	[0] 2	ĸ		Fanout	ine s_typi ells ibrary
-	ple.Re TMR_S1			25.5	8.8	110 1	7.7	15.4	15.4		13.3	80.1	8.8	13.4			Load (ff)	cal re
	ps[0].	1	•	772	DEC	1410	103	202	499		155	117	383	71			(ps)	vision
	Reg_uni egs[2]		145 a	485 8	+111 	-700 -700	\$ ±	+172 +8	100 to		±379	+183	+191	+249			Oelay /	1.0
	Reg_units	12068 R	8872 9017 R	8387 8872 R	8275 8756	7496	7496 R	7228 7460 F	5923 7228 R		6544 6922 F	6544 F	6449 R	9249 F	6000 F	6000 F	(ps)	
	Dout_reg/D																	

Σχήμα 3.34: Στατιστικά για το κρίσιμο μονοπάτι σε απλό TMR για υλοποίηση: (α) με τα τροποποιημένα κελιά. (β) με τα πρωτότυπα κελιά.

(a)	<pre>.clk') ./TMP_Simple.Regs[0].Reg_units/Dout_reg[0]/CP .Lum .t_regs[2].Shift_register/Shift_reg[0].Shift_register/Dout_reg/SE Star End-</pre>	12800 R (clo	Sh +0 8429 	20 21 21	24 258.9 2074 +1889 8429 R FSM/	2 22.8 289 +172 7348 R gl	2 19-5 186 +174 7167 R 91 +6 7167 91	4 JIL JOS - JUN 400 994 91	10 B++ 007 1 FE A	92 92	2 28.6 129 +244 6656 F FSM/	1 12.9 117 +91 6412 F TM +0 6412 Take	4 510 - 512	a ices sait acc a ri i	[0] [0]1	2 20.7 65 +155 6156 F TN	14	5498 F	Fanout Load Slew Delay Arrival (clo (fF) (ps) (ps) take	Library
	Group : 'clk' (path_group 'cl) ng slack : 374ps t.point : take_addr_state_reg/T point : data_shift_reg/Shift_)	ck clk) capture	<pre>ift_regs[1].Shift_register/en Shift_reg[8].Shift_register/en Dout_reg/SE <<< DFCS_X1 Dout_reg/CP setup</pre>	_shift_reg/en3 //A BUF_X1	en_8bit	416/BL AD31 Y845	438/Q AMD2_X8d5	A32/Q NOR2_X8d5	454/0 INV_X1	/Q NOR2_X8d5	CurrState[0]	<pre>#_Simple.voter_unit/voter_out[8] addr_state_reg/Data_out1[8]</pre>	g130/0 A021_X0d5	g134/0 ADIZIZ805	g141/0 INV_X1 g134/A2	R_Simple.voter_unit/voter_in1(0) g141/A	Dout_reg[8]/Q DFWC_X1 R Simple.Reds[8].Red units/Dout[4	Dout_reg(0)/CP	ck clk) lawnch	Pin Type I
(B)	(') MSimple.Regs[0].Reg_units/Dout regs[1].Shift_register/Shift_reg	12000	-0 11093 0 +533 11626	+0 18476 8 83.7 844 +623 11893	av avert sant tanta avert	10 187 7 7177 1741 18478	+0 8/29 2 22.3 501 +469 9228	1 12.7 548 +425 8759	2 21.5 354 +294 8332 +0 8333	3 29.9 1360 +790 8038 +0 8039	87.0.6		2 20.3 260 +593 7248	1 12.8 226 +176 6654	1 13.8 174 +133 6478 +0 6478	+0 6345	2 22.3 133 +345 6345 91	0 6000	6499	Fanout Load Slew Delay Arrival (fF) (ps) (ps) (ps)

Σχήμα 3.35: Στατιστικά για το κρίσιμο μονοπάτι σε απλό TMR για υλοποίηση με τα τροποποιημένα κελιά: (α) σε ακρότατο σημείο FF. (β) σε ακρότατο σημείο SS.

Στον πίνακα 8 παρουσιάζονται τα στατιστικά για την κατανάλωση ενέργειας για υλοποιήσεις με τροποποιημένα και πρωτότυπα κελιά για ακρότατο σημείο TT.

Πίνακας 8: Στατιστικά κατανάλωσης ισχύος σε απλό TMR για υλοποίηση με τροποποιημένα και πρωτότυπα κελιά.

Υλοποίηση	Εσωτερική ισχύς	Ισχύς μεταγωγής	Συνολική δυναμική ισχύς	Ισχύς διαρροής
Με τροποποιημένο κελιά	0,36 mW	0,37 mW	0,73 mW	0,016 pW
Με πρωτότυπα κελιά	0,51 mW	0,42 mW	0,93 mW	0,011 pW

Στον πίνακα 9 παρουσιάζονται τα στατιστικά για την κατανάλωση ενέργειας για υλοποίηση με τροποποιημένα κελιά για ακρότατα σημεία FF και SS.

Πίνακας 9: Στατιστικά κατανάλωσης ισχύος σε απλό TMR για ακρότατα σημεία FF και SS.

Ακρότατο σημείο	Εσωτερική ισχύς	Ισχύς μεταγωγής	Συνολική δυναμική ισχύς	Ισχύς διαρροής
FF	0,38 mW	0,37 mW	0,75 mW	0,014 pW
SS	0,33 mW	0,36 mW	0,69 mW	0,017 pW

3.5.2 Αποτελέσματα για την υλοποίηση με χρήση πλήρους TMR

Για την υλοποίηση με πλήρες TMR συνολικά χρησιμοποιήθηκαν 909 κελιά με συνολική επιφάνεια 30.318 μm2 για την υλοποίηση με τα τροποποιημένα κελιά, ενώ για την υλοποίηση με τα πρωτότυπα κελιά χρησιμοποιήθηκαν 912 κελιά με συνολική επιφάνεια 15.204 μm2. Στο σχήμα 3.36 (α) βλέπουμε τα στατιστικά για την υλοποίηση με δακτυλίους προστασίας, ενώ στο σχήμα 3.36 (β) για την υλοποίηση με τα πρωτότυπα κελιά. Στον αριθμό των buffer δεν περιλαμβάνονται οι buffers που χρησιμοποιήθηκαν για τα δέντρα των ρολογιών. Οι αλλαγές στον αριθμό των κελιών που χρησιμοποιήθηκαν οφείλεται στο ότι η βιβλιοθήκη έχει διαφορετικούς χρόνους καθυστέρησης, κατανάλωσης ισχύος κλπ. Έτσι το εργαλείο σύνθεσης επέλεξε τη βέλτιστη περίπτωση. Η συνολική επιφάνεια είναι διπλάσια, διότι το κάθε τροποποιημένο κελί είναι διπλάσιο από το αντίστοιχο πρωτότυπο.

the second secon	the second second second second second				<u> </u>		
Generated Generated Module: Technolog Operating Intercond Area mode	d by: d on: gy library: g conditions: nect mode: e:	Genus(TM Sep 83 2 spi_slav lvBhvt_g PVT_lP8v global physical	1) Synthesis Solution GENUS15.28 1016 B8:22:07 am 10_full_TMR 10_ardring_tt_1.8V_25 1.8 (_25C 11brary	Generate Generate Module: Technolo Operatin Intercon Area mod	d by: d on: gy libraries: g conditions: nect mode: e:	Genus(17 Feb 07 ; spi_slan lf15adhu physical typical global physical	M) Synthesis Solution GENU515.20 2016 06:27:06 am ve_full_TMR rfs_typical revision 1.0 L_cells L library
Gate	Instances	Area	Library	Gate	Instances	Area	Library
AND2_X0d5	75	1807.920	1v8hvt_guardring_tt_1.8V_25			*********	
A021_X0d5	144	4339.088	lv8hvt guardring tt_1.8V_25	AND2_X8d5	81	976.277	lf15adhvt9s_typical
A0121_X8d5	21	506.218	1v8hvt_guardring_tt_1.8V_25	A021_X8d5	147	2214.702	lf15adhvt9s_typical
CLKBUF_X1	42	759.326	1v8hvt_guardring_tt_1.8V_25	A0121_X0d5	18	216.958	lf15adhvt9s_typical
DFCS_X1	36	5423.760	1v8hvt_guardring_tt_1.8V_25	CLKBUF_X1	- 42	379.663	lf15adbvt9s_typical
DFC_X1	21	2657.642	1v8hvt_guardring_tt_1.8V_25	DFCS_X1	36	2711.898	lf15adhvt9s_typical
DFNC_X1	57	6183.086	1v8hvt_guardring_tt_1.8V_25	DFC_X1	21	1328.821	lf15adhvt9s_typical
HA_X1	6	361.584	1v8hvt_guardring_tt_1.8V_25	DFNC_X1	57	3091.543	if15adhvt9s_typical
INV_X1	189	2277.979	1v8hvt_guardring_tt_1.8V_25	HA_X1	5	180.792	if15adhvt9s_typ1cal
NOR2_X0d5	386	5532.235	1v8hvt_guardring_tt_1.8V_25	INV_XL	192	1157.009	tfloadhvt9s_typical
0A122_X8d5	6	188.792	1v8hvt_guardring_tt_1.8V_25	NOR2_X8d5	388	2711.888	lf15adhvt9s_typical
X0R2_X1	6	289.267	1v8hvt_guardring_tt_1.8V_25	0AI22_X0d5	6	90,396	lf15adhvt9s_typical
+ + + + + + + + + + + + + + + + + + + +			**********	X082_X1	0	144.034	LT15adhvt9s_typical
total	989	38318.818		total	912	15284.687	
Туре	Instances	Area Are	a %	Type	Instances	Area Are	
requests al	114 14	1264 400 4	7.0	. 1910			
investor	114 14	204.409 4	7.0	sequential	114 7	132.244	46.9
Ander Ler	109 /	750 326	3.5	inverter	192 1	157.069	7.6
Janie	564 12	017 024 A	2.J	buffer	42	379,663	2.5
cours	304 13	OTI-OTA 4	141+18	logic	564 6	535.631	43.0
total	000 30	319 919 10	10 B	*********			
totat	303 30	1210-010 10	50° + 60	total	912 15	204.687 10	90.0
		(0	χ)				(β)

Σχήμα 3.36: Στατιστικά για τα κελιά που χρησιμοποιήθηκαν και για την συνολική επιφάνεια σε πλήρες TMR για υλοποίηση: (α) με τα τροποποιημένα κελιά. (β) με τα πρωτότυπα κελιά.

Dubut_reg(1)/0 Diver_(x)1 6 33.7 128 +287 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 NOR2_X0d5 1 8.1 117 +104 6594 F g133/A1 A021_X0d5 2 13.2 13.2 155 +379 6972 F g133/A1 A021_X0d5 2 13.2 155 +379 6972 F g133/A1 A021_X0d5 2 13.2 155 +379 6972 F g131/A1 A021_X0d5 2 13.2 155 +379 6972 F g131/A1 A021_X0d5 2 13.2 155 +379 6972 F g1421/A1 A072_X0d5 3 20.9 615 +363 7336 R 1441/A2 NOR2_X0d5 7 10 +10 7537 F 14421/A1 AND2_X0d5 7 40.9 548 +360 8008 R 1441/B1 H21/A1 AND2_X0d5 7 38.9 1114 +630 8722 R <th></th> <th></th> <th>(3) capture</th>			(3) capture
Dout_reg(1)/0 DFWC_X1 6 33.7 128 +287 6287 gl36/A2 NOR2_X0d5 1 8.3 305 +203 6490 R gl36/A2 NOR2_X0d5 1 8.1 117 +104 6594 F gl33/A1 NOR2_X0d5 1 8.1 117 +104 6594 F gl33/A1 A021_X0d5 2 13.2 13.5 +379 6972 F gl33/A1 A021_X0d5 2 13.2 13.5 +379 6972 F gl31/B1 A021_X0d5 2 13.2 13.5 +379 6972 F gl31/A1 A021_X0d5 2 16.4 235 +201 7537 F gl35/A2 NOR2_X0d5 7 40.9 548 +40 7336 F 1441/A2 NOR2_X0d5 7 40.9 548 +360 8000 F 1411/B1 H49 A0121_X0d5 1 <th></th> <th></th> <th></th>			
Dout_reg(1)/0 DFWC_X1 6 33.7 128 +287 6287 IMP_Full.Regs[0].Reg_units/Dout[1] +0 63.7 128 +287 6287 g136/A2 NOR2_X0d5 1 8.1 117 +104 6594 g133/A1 G12.X0d5 2 13.2 155 +203 6490 R g133/A1 G12.X0d5 2 13.2 155 +379 6972 F g131/B1 4021_X0d5 2 13.2 155 +379 6972 F g131/A1 A021_X0d5 2 13.2 155 +379 6972 F g131/B1 4021_X0d5 2 12.5 +379 6972 F g135/A2 NOR2_X0d5 3 20.9 615 +363 7336 g141/A2 NOR2_X0d5 3 20.9 615 +363 7336 1441/A2 NOR2_X0d5 1 17.7 110 +0 7537 F 1435/A NND2_X0d5 7 40.9 548 +360 8000 R	10475 R	0 +133	t_reg/CP setup
Dout_reg[1]/0 DHVC_X1 6 33.7 128 +287 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 6491 6491 6491 6491 <td>10342</td> <td>+0</td> <td>c_reg/D <<< DFC_X1</td>	10342	+0	c_reg/D <<< DFC_X1
Dout_reg[1]/0 DHVC_X1 6 33.7 128 +287 6287 gl35/A2 NOR2_X0d5 1 8.8 305 +203 6490 6491 6491 6491 6493 640 7336			<pre>//ll.Regs[2].Reg_units/Din</pre>
Dout_reg[1]/0 DHNC_X1 6 33.7 128 +287 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 NOR2_X0d5 1 8.1 117 +10 6594 F g133/A1 NOR2_X0d5 2 13.2 155 +379 6972 F g131/0 A021_X0d5 2 13.2 155 +379 6972 F g131/0 A021_X0d5 2 13.2 155 +379 6972 F g135/0 A021_X0d5 2 13.2 155 +379 6972 F g135/0 A021_X0d5 2 16.1 17.7 146 6594 F g145/0 INV_X1 1 7.7 110 +10 7336 R 1441/0 INV_X1 1 7.7 110 +10 7640 R 1421/0 INV_X1 1 8.8 133 +92 8000 R 1411/81 INV_X1 1 8.8	0		reg/Data_in3
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 gl36/A2 NOR2_X0d5 1 8.8 305 +203 6490 R gl36/A2 NOR2_X0d5 1 8.1 11.7 +104 6594 F gl33/A1 NOR2_X0d5 1 8.1 11.7 +104 6594 F gl33/A1 NOR2_X0d5 2 13.2 155 +379 6972 F gl31/B1 A021_X0d5 2 13.2 155 +379 6972 F gl31/PI A021_X0d5 2 13.2 155 +379 6972 F gl31/PI A021_X0d5 2 16.4 235 +379 6972 F gl32/PI MR_Full/Voter_unit/voter_out[1] H0 515 +363 7336 R gl32/PI NOR2_X0d5 3 20.9 615 +363 7336 R 1441/PI NOR2_X0d5 7 40.9 <td>add</td> <td></td> <td>unter/en3</td>	add		unter/en3
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 gl36/A2 NOR2_X0d5 1 8.8 305 +203 6490 R gl36/A2 NOR2_X0d5 1 8.1 117 +10 6594 F gl33/A1 NOR2_X0d5 2 13.2 155 +379 6972 F gl33/A1 NOR2_X0d5 2 13.2 155 +379 6972 F gl31/0 NOR2_X0d5 2 13.2 155 +379 6972 F gl31/0 NOR2_X0d5 2 13.2 155 +379 6972 F gl31/0 NOR2_X0d5 2 16.4 235 +379 6973 F gl35/0 NOR2_X0d5 2 16.4 235 +201 7537 F 1441/0 NOR2_X0d5 7 40.9 548 +360 8008 R 1441/0 INV_X1 1 8.8 <td< td=""><td>FSM</td><td></td><td>haddr cntr voter/voter out</td></td<>	FSM		haddr cntr voter/voter out
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 F g136/A2 NOR2_X0d5 1 8.1 11.7 +40 639.7 6287 F g136/A2 NOR2_X0d5 1 8.1 11.7 +40 639.7 6287 F g133/A1 NOR2_X0d5 1 8.1 11.7 +104 6594 F g131/0 NOR2_X0d5 2 13.2 155 +379 6972 F g131/0 A021_X0d5 2 13.2 155 +379 6972 F g135/0 A021_X0d5 2 13.2 155 +379 6972 F g135/0 A021_X0d5 2 16.4 235 +363 7336 R g145/0 NOR2_X0d5 2 16.4 235 +201 7537 F 1441/0 NOR2_X0d5 7 40.9 548 +360 8000 R 1412/0 <t< td=""><td>10342 R g</td><td>+0 13.8 226 +239</td><td>A021 X0d5</td></t<>	10342 R g	+0 13.8 226 +239	A021 X0d5
Dout_reg[1]/0 DMR_UIL.Regs[0].Reg_units/Dout[1] 6 33.7 128 +287 6287 g136/A2 NOR2_X0d5 1 8.1 11.7 +10 639.7 g135/A2 NOR2_X0d5 1 8.1 11.7 +10 639.7 g133/A1 NOR2_X0d5 1 8.1 11.7 +10 6594 g133/A1 NOR2_X0d5 2 13.2 155 +379 6972 g131/0 A021_X0d5 2 13.2 155 +379 6972 g131/0 A021_X0d5 2 13.2 155 +379 6972 g131/0 A021_X0d5 2 13.2 155 +379 6972 g135/0 NOR2_X0d5 3 20.9 615 +363 7336 g145/0 NOR2_X0d5 2 16.4 235 +201 7537 1441/0 INV_X1 1 7.40.9 548 +360 8000 1421/0 INV_X1 1		00CT 07H 0.77	CDBV_7UON
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 F gl36/A2 NOR2_X0d5 1 8.8 305 +203 6490 R gl36/A2 NOR2_X0d5 1 8.1 117 +104 6594 F gl33/A1 NOR2_X0d5 1 8.1 117 +104 6594 F gl33/A1 NOR2_X0d5 2 13.2 155 +379 6972 F gl31/0 A021_X0d5 2 16.4 235 +379 6973 F gl35/0 NOR2_X0d5 7 40.9 548 +40 7336 F 1435/0 INV_X1 1 7.	5 /6/6	900' 00' 0 CI	
Dout_reg[1]/0 DHVC_X1 6 33.7 128 +287 6287 PMF_full.Voters[0].voter_unit/voter_in1[1] 9136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g135/A2 NOR2_X0d5 1 8.1 117 +104 6594 F g133/A1 NOR2_X0d5 2 13.2 155 +379 6972 F g131/0 A021_X0d5 2 13.2 155 +379 6972 F g135/0 NOR2_X0d5 3 20.9 615 +363 736 F g145/0 NOR2_X0d5 2 16.4 235 +201 7537 F g1421/A1 H	9796 F g	. 13.9 381 +258	NOR2_X0d5
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 F gl36/A2 NOR2_N0d5 1 8.8 305 +203 6490 R gl36/A2 NOR2_X0d5 1 8.1 117 +10 6594 F gl33/A1 NOR2_X0d5 1 8.1 117 +104 6594 F gl33/A1 NOR2_X0d5 2 13.2 155 +379 6972 F gl31/0 A021_X0d5 2 16.4 215 +363 7336 F gl35/A1 NOR2_X0d5 2 16.4 235 +201 7537 F gl450/A INV_X1 1	9538 g	1 0+	
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 F gl36/A2 NOR2_N0d5 1 8.8 305 +203 6490 R 6490 R 9135/A1 +0 6287 6287 9136/A2 NOR2_N0d5 1 8.1 117 +104 6594 6490 R 9131/A1 +0 6594 631.7 117 +104 6594 F 9131/O A021_X0d5 2 13.2 155 +379 6972 F 6397 F 6397 F 6397 F 631.7 118 117 +104 6594 F 9131/O A021_X0d5 2 13.2 155 +379 6972 F 6 6193 1417 140 6594 F 615 1417 140 653 7336 R +0 1336 R 14141/A2 +0 7336 R 1435/A +0 7336 7 1435/A 10 +13	MS-		addr_cntr_voter/voter_in1
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 MR_Full.Regs[0].voter_unit/voter_in1[1] 9136/A2 +0 6287 -0 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 NOR2_X0d5 1 8.1 117 +104 6594 F g133/A1 NOR2_X0d5 2 13.2 155 +379 6972 F g131/0 A021_X0d5 2 16.1 17.7 160 5973 F g145/A2 NOR2_X0d5 2 16.4 235 +201 7537 F 1458/A INV_X1 1 7.7 110		U.1 1010 TO/O	SM/en addr ont
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] 6 33.7 128 +287 6287 F g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 NOR2_X0d5 1 8.1 117 +10 6594 F g131/0 A021_X0d5 2 13.2 155 +379 6972 F g134/0 A021_X0d5 2 13.2 155 +379 6972 F g145/A2 MR_eull.Voters[0] N0R2_X0d5 3 20.9 615 +363 7336 R 1441/A2 N0R2_X0d5	00000	01 27 1 1201 1070	
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] 40 633.7 128 +40 6287 6287 6287 6287 633.7 128 +40 6287 6490 6133/0 N0R2_X0d5 1 8.8 305 +203 6490 6490 6133/0 40 6594 6391 613 117 +104 6594 6391 631 117 +104 6594 6391 631 117 +104 6594 6391 631 117 +104 6594 6391 631 117 +104 6594 6391 631 117 +104 6594 631 631 117 +104 6594 631 631 117 +104 6594 631 132 115 +353 6372 6972 6972 6973 1458 40 1336 8 141/4 40 1336 7336 7336 7336 <t< td=""><td>8660 F g</td><td>. 13.8 190 +131</td><td>IX_ANI</td></t<>	8660 F g	. 13.8 190 +131	IX_ANI
Dout_reg[1]/0 DHV_X1 6 33.7 128 +287 6287 MR_Full.Regs[0].Reg_units/Dout[1] 9136/A2 +0 6287 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 g136/A2 NOR2_X0d5 1 8.1 117 +104 6594 g133/A1 NOR2_X0d5 2 13.2 155 +379 6972 g131/B1 A021_X0d5 2 13.2 155 +379 6972 g131/P A021_X0d5 2 15.5 +379 6973 g141/A2 NOR2_X0d5 3 20.9 615 +66 1441/A2 NOR2_X0d5 2 16.4 235 +20 7336 </td <td>g 258</td> <td>÷</td> <td></td>	g 258	÷	
Dout_reg[1]/0 DHVC_X1 6 33.7 128 +287 6287 F MR_Full.voters[0].voter_unit/voter_in1[1] 9136/A2 +0 6287 6287 6287 6287 6287 6287 6287 633.7 1.28 +0 6287 633.7 1.28 +0 6287 633.7 1.28 40 6287 6394 6393 6394 6391 +0 6594 6397 637 637 637 637 637 6373 6373 6373 6373 6373 6373 6373 6373 6373 6373 6373 64973 1441/A2 +0 7336 7336 6441/A2 +0 7336 6441/A2 +0 7336 441/A3 <td>8529 R g</td> <td>59.3 779 +478</td> <td>AND2_X0d5</td>	8529 R g	59.3 779 +478	AND2_X0d5
Dout_reg[1]/0 DMR_Z 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] 6 33.7 128 +287 6287 6287 6287 6287 6287 6287 633.7 128 40 6287 633.7 128 +0 6287 633.7 128 40 6287 6398 6393 633.7 128 40 6390 633.7 128 40 6390 6393 6393 6190 633.7 117 +0 6394 6390 633.7 633 6394 6391 6394 633.7 6394 6394 633.7 633 6394 633.7 633 615 +363 6394 6394 633.7 6496 633.7 637 6496 633.7 6972 6 6773 6972 6 6773 6973 6973 6973 6973 6973 6973 6973 6 6973 6 6973 6 6537 <td< td=""><td>8051 g</td><td>,</td><td>-</td></td<>	8051 g	,	-
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] g136/A2 +0 6287 F g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 +0 6287 g133/A1 NOR2_X0d5 1 8.1 117 +104 6594 F g131/B1 A021_X0d5 2 13.2 155 +379 6972 F g131/V0ters[0].voter_unit/voter_out[1] e_addr_state_reg/Data_out1[1] s[0].FSM/CurrState[1] 1458/A2 NOR2_X0d5 3 20.9 615 +363 7336 R 1441/A2 NOR2_X0d5 2 16.4 235 +201 7537 F 1458/A +0 7537 F	8050 R g	2 19.5 186 +164	IX VNI
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] 1 1 6 33.7 128 +287 6287 6287 6287 6287 6287 6287 6287 633.7 1 8.8 305 +203 6490 8 9135/02 1 8.1 117 +104 6594 F 6133/0 6393 7 1 16 6594 F 6133/0 N0R2_X0d5 1 8.1 117 +104 6594 F 6131/0 61394 F 61394 F 6131/0 6131/0 6131/0 1001/0 6101/0 6101/0 6101/0 6101/0 6101/0 6101/0 6101/0 6101/0 6101/0 6173 6101/0 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 6173 61737	7886	0+ 012: 100 0101	
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] g136/A2 +0 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 NOR2_X0d5 1 8.1 117 +104 6594 F g131/B1 A021_X0d5 2 13.2 155 +379 6972 F g131/B1 A021_X0d5 2 13.2 155 +379 6972 F MR_Full.Voters[0].voter_unit/voter_out[1] s[0].FSM/CurrState[1] +0 6973 1458/A2 +0 6973 3 20.9 615 +363 7336 R	7886 F	856+ 705 8 51	NORS XD45
Dout_reg[1]/0 DMV_X1 6 33.7 128 +287 6287 MR_Full.Regs[0].Reg_units/Dout[1] 6 33.7 128 +287 6287 MR_Full.Noters[0].voter_unit/voter_in1[1] 9136/A2 +0 6287 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 g133/A1 NOR2_X0d5 1 8.1 117 +104 6594 g131/B1 NOR2_X0d5 1 8.1 117 +104 6594 g131/B1 NOR2_X0d5 2 13.2 155 +379 6972 g131/0 NOR21_X0d5 2 13.2 155 +379 6972 g131/0 A021_X0d5 2 13.2 155 +379 6972 g138/0 A021_X0d5 2 13.2 155 +379 6972 g138/0 A021_X0d5 2 13.2 155 +379 6972 g148/A2 Yoters[0].voter_unit/voter_out[1] +0	7647 9	01CT /TOT C.OC 1	Notice
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] 6 33.7 128 +287 6287 F gl36/A2 NOR2_X0d5 1 8.8 305 +203 6490 R gl33/A1 NOR2_X0d5 1 8.1 117 +104 6490 R gl33/A1 NOR2_X0d5 1 8.1 117 +104 6594 F gl31/B1 NOR2_X0d5 1 8.1 117 +104 6594 F gl31/B1 A021_X0d5 2 13.2 155 +379 6972 F mR_Full.Voters[0].voter_unit/voter_out[1] e_addr_state_reg/Data_out1[1] s[0].FSM/CurrState[1]	76/7 B 0	0751 7101 0 36 1 0+	NOB3 YEAS
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] g136/A2 +0 6287 g135/A1 +0 6287 g133/A1 +0 6287 g133/A1 +0 6287 g133/A1 +0 6287 g133/A1 +0 6594 F g133/A1 NOR2_X0d5 1 8.1 117 +164 6594 F g131/B1 +0	MS-	5	om/currstate[1]
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_full.Voters[0].voter_unit/voter_in1[1] g136/A2 +0 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 +0 6490 g133/A1 NOR2_X0d5 1 8.1 117 +164 6594 F g131/0 NOR2_X0d5 1 8.1 117 +164 6594 F g131/0 +0 6594 F	tak		state_reg/Data_out1[1]
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_Full.Voters[0].voter_unit/voter_in1[1] +0 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 +0 6490 g133/A1 +0 6490 g133/A1 +104 6594 F g131/B1 +0 6594 F g131/Q A021_X0d5 2 13.2 155 +379 6972 F	1	Ξ	.Voters[0].voter_unit/voter_out
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_Full.Voters[0].voter_unit/voter_in1[1] +0 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 +0 6490 g133/A1 +0 6490 g133/A1 +0 6594 F g133/A1 +0 6594 F g133/A1 +0 6594 F	7077 F	20.5 180 +376	A021_X0d5
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_Full.Voters[0].voter_unit/voter_in1[1] +0 6287 g136/A2 +0 6287 g136/A2 NOR2_X0d5 1 8.8 305 +203 6490 R g133/A1 +0 6490 R g133/A1 NOR2_X0d5 1 8.1 117 +104 6594 F	6701	+0	
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_Full.Voters[0].voter_unit/voter_in1[1] +0 6287 gl36/A2 +0 6287 gl36/Q NOR2_X0d5 1 8.8 305 +203 6490 R	6700 F	. 12.9 169 +147	NOR2 X0d5
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_Full.Voters[0].voter_unit/voter_in1[1] 		0121 2CF C.CT -	Noix_vous
Dout_reg[1]/0 DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1] MR_Full.Voters[0].voter_unit/voter_in1[1]	6553 D	9467 66V 0 61	
Dout_reg[1]/Q DFNC_X1 6 33.7 128 +287 6287 F MR_Full.Regs[0].Reg_units/Dout[1]	T Trees	1	Voters[0].voter_unit/voter_in1
Dout_reg[1]/Q DFNC_X1 6 33.7 128 +287 6287 F	Т		Regs[0].Reg_units/Dout[1]
	6277 F	6 49.6 171 +277	reg[1]/Q DFNC_X1 (
Dout reg[1]/CP 0 6000 F	6000 F	o	reg[1]/CP
MR Full.Reas[0].Rea units	L		[.Regs[0].Reg units
ייסקעי גראדי בשווכוו סמממ ב			state ren soundi

Σχήμα 3.37: Στατιστικά για το κρίσιμο μονοπάτι σε πλήρες TMR: (α) για υλοποίηση με τα τροποποιημένα κελιά. (β) για υλοποίηση με τα πρωτότυπα κελιά.

(α)

(β)

				(B)							(a)		
1]/CP .Shift_register/Dout_reg/SF	3/Dout_reg[hift_reg[8]	eg_unit ister/5	s (0).Re ft_regi	<pre>'HR_Full.Regs regs[2].Shif</pre>	shift_reg/Shift_ shift_reg/Shift_	Timing slack : 1 Timing slack : 1 Start-point : take End-point : data	g[1]/CP g[4].Shift_register/Dout_reg/SE	its/Dout_ /Shift_re	Reg_un igister	gs[0]. 1ft_re	_regs[2].Sh	addr_state_reg/ shift_reg/Shift	Timing slack : 34 Start-point : take End-point : addr
	12080 R				capture	(clock clk3)		12890 R				capture	(clock clk3)
	11323 11897 R	-574 -574	a	, , , ,	er/vocer_ous ft_register/en Lft_register/en setup setup	data_shift_reg(e1) shift_reg(2).Shi Shift_reg(3).Shi Bout_reg/SE - Dout_reg/CP		8389 8520 R	+210		* * * *	rr/volei_out t_register/en cec DFCS_X1 setup	addr_shift_regray_vor shift_regs[2].Shif Shift_reg[4].Shi Dout_reg/SE Dout_reg/CP
	10323 R	866+	1742	8 83.7	A021_X0d5	g48/A1 g48/0		8302 R	+282	488	4 46.9	A021_X8d5	0/660 18/0
	18132 18325 R	+193	284	1 12.8	IN/_X1	g3/A g3/Q		7837 8027 R	*198 +8	274	1 12.5	NOR2_X8d5	g34/AL g34/Q
	10131 F	+337	343	1 15.6	NOR2_X8d5	92/0		7837 F	+143	236	1 14.0	NOR2_X8d5	g35/0
	0704	ŧ			er/woter_inl	FSMb[0].FSM/en_Bbit_vote		7694	\$			er/voter_inl	FSMs[0].FSMVen_4011 FSMs[2].en_4bit_vote
	99990 9794 R	+708	1035	6 48.7	A021_X0d5	g1494/81 g1494/0		7694 R	40 đ	25	6 49.9	NDR2_X8d5	g1353/0
	8 9886 C/00	411	501	2 22.3	AND2_X8d5	g1448/0 g1448/0		7210 F	+167	228	2 21.9	NOR2_X8d5	91370/0
	8675 R	+467	319	1 12.7	AND2_X8d5	g1451/0		7843 R	*338	599	4 34, 3	NOR2_X8d5	g1384/0
	7278 8288 R	+938	1558	4 34.6	MUR2_X0d5	g1493/A2 g1493/Q		6-70E			er_out[1]	<pre>//voter_unit/vot /Data_out1[1] te[1]</pre>	FSMs[0].FSM/CurrStat
				r_out[1]].voter_unit/vote /Data_out1[1]	TMM_Full.Voters[8 take_addr_state_reg,		6462 6785 F	+244	129	2 20.6	A021_X8d5	g131/81 g131/81
	6373 7278 F	+498	428	2 28.3	0AI22 X8d5	0131/A1		6370 6461 F	\$ ±	118	1 12.9	NOR2 X8d5	g133/A1
	6426 6878 R	±452	688	1 13.8	MOR2_X0d5	g133/A2		6193 6378 R	*176	278	1 14.6	NOR2 X8d5	0136/A2
	0440 1	0744	200	Line a	Reg_units/Dout[1]	TMR_Full.Regs[0].F		1.0000	100	42.0]	Meg_units/Dout[1	TMR Full.Regs[8].R
	6000 F	206	5	n h	Reg_units	TMR_Full.Regs[0].F Dout_reg[1]/CP		6000 F	103		h D	leg_units	TMR_Full.Regs(8).P Dout_reg[1]/CP
	6969 F				Launch	(clock clk1)		6000 F				launch	(clock clk1)
	(ps)	(ps)	(ps)	Fanout Load (fF)	Туре	Pin		Arrival (ps)	(ps)	(ps)	Fanout Loac (fF)	Type	PLn

Σχήμα 3.38: Στατιστικά για το κρίσιμο μονοπάτι σε πλήρες TMR για υλοποίηση με τα τροποποιημένα κελιά: (α) σε ακρότατο σημείο FF. (β) σε ακρότατο σημείο SS.

Στην εικόνα 3.37 πιο πάνω βλέπουμε το κρίσιμο μονοπάτι, δηλαδή το πιο αργό κομμάτι του κυκλώματος, (α) για υλοποίηση με τα τροποποιημένα κελιά και (β) για υλοποίηση με τα πρωτότυπα κελιά.

Και στις δύο περιπτώσεις το κρίσιμο μονοπάτι είναι το ίδιο. Ξεκινάει από το ρολόι του καταχωρητή κατάστασης για την αποθήκευση της διεύθυνσης και καταλήγει στην είσοδο του καταχωρητή του μετρητή για τα bit διεύθυνσης. Για την υλοποίηση με τα τροποποιημένα κελιά το κρίσιμο μονοπάτι είναι 4,475 ns, ενώ για την υλοποίηση με τα πρωτότυπα κελία το κρίσιμο μονοπάτι είναι 3,406 ns. Ο καταχωρητής κατάστασης λειτουργεί στην καθοδική αιχμή του ρολογιού, οπότε το κρίσιμο μονοπάτι ξεκινάει από τα 6 ns και ολοκληρώνεται στα 10,475 ns για υλοποίηση με τα τροποποιημένα κελιά και 9,406 ns για υλοποίηση με τα πρωτότυπα κελιά. Απομένουν 1,525 ns και 2,594 ns για τις δύο υλοποιήσεις αντίστοιχα μέχρι την ολοκλήρωση της περιόδου του παλμού του ρολογιού και την έλευση της θετικής ανόδου που λειτουργεί ο καταχωρητής ολίσθησης δεδομένων.

Στην εικόνα 3.38 πιο πάνω βλέπουμε το κρίσιμο μονοπάτι για υλοποίηση με τα τροποποιημένα κελιά, (α) για ακρότατο σημείο FF στους -40° C και (β) για ακρότατο σημείο SS στους 70° C.

Για το σημείο FF το κρίσιμο μονοπάτι είναι 2,520 ns. Ξεκινάει από το ρολόι του καταχωρητή κατάστασης για την αποθήκευση της διεύθυνσης και καταλήγει στην είσοδο του σήματος ενεργοποίησης του καταχωρητή ολίσθησης διευθύνσεων. Ο καταχωρητής κατάστασης λειτουργεί στην καθοδική αιχμή του ρολογιού, οπότε το κρίσιμο μονοπάτι ξεκινάει από τα 6 ns και ολοκληρώνεται στα 8,520 ns και απομένουν 3,480 ns μέχρι την ολοκλήρωση της περιόδου του παλμού του ρολογιού και την έλευση της θετικής ανόδου που λειτουργεί ο καταχωρητής ολίσθησης διευθύνσεων.

Για το σημείο SS το κρίσιμο μονοπάτι είναι 5,897 ns. Ξεκινάει από το ρολόι του καταχωρητή κατάστασης για την αποθήκευση της διεύθυνσης και καταλήγει στην είσοδο του σήματος ενεργοποίησης του καταχωρητή ολίσθησης δεδομένων. Ο καταχωρητής κατάστασης λειτουργεί στην καθοδική αιχμή του ρολογιού, οπότε το κρίσιμο μονοπάτι ξεκινάει από τα 6 ns και ολοκληρώνεται στα 11,897 ns και απομένουν 0,103 ns μέχρι την ολοκλήρωση της περιόδου του παλμού του ρολογιού και την έλευση της θετικής ανόδου που λειτουργεί ο καταχωρητής ολίσθησης δεδομένων.

Στον πίνακα 10 παρουσιάζονται τα στατιστικά για την κατανάλωση ενέργειας για υλοποιήσεις με τροποποιημένα και πρωτότυπα κελιά για ακρότατο σημείο TT.

Πίνακας 10: Στατιστικά κατανάλωσης ισχύος σε πλήρες TMR για υλοποίηση με τροποποιημένα και πρωτότυπα κελιά.

Υλοποίηση	Εσωτερική ισχύς	Ισχύς μεταγωγής	Συνολική δυναμική ισχύς	Ισχύς διαρροής
Με τροποποιημένο κελιά	0,47 mW	0,75 mW	1,22 mW	0,023 pW
Με πρωτότυπα κελιά	0,55 mW	0,50 mW	1,05 mW	0,016 pW

Στον πίνακα 11 παρουσιάζονται τα στατιστικά για την κατανάλωση ενέργειας για υλοποίηση με τροποποιημένα κελιά για ακρότατα σημεία FF και SS.

Ακρότατο σημείο	Εσωτερική ισχύς	Ισχύς μεταγωγής	Συνολική δυναμική ισχύς	Ισχύς διαρροής
FF	0,50 mW	0,76 mW	1,26 mW	0,022 pW
SS	0,43 mW	0,74 mW	1,17 mW	0,025 pW

4. ΣΥΝΟΨΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτή την εργασία έγινε ο χαρακτηρισμός σε τροποποιημένα πρότυπα κελιά, στα οποία τροποποιήθηκε η φυσική τους σχεδίαση και προστέθηκαν δακτύλιοι προστασίας ώστε να μειωθεί η πιθανότητα ανοικτοκλειδώματος.

Δημιουργήσαμε βιβλιοθήκη κελιών κατάλληλη για χρήση από εργαλείο σύνθεσης ψηφιακών κυκλωμάτων. Υλοποιήσαμε δύο εκδοχές του υποτελούς (slave) του σειριακού πρωτόκολλου SPI, με σκοπό την ανθεκτικότητα σε συμβάντα μεμονωμένης διέγερσης, τα οποία προκαλούνται από τα φορτισμένα σωματίδια της κοσμικής ακτινοβολίας στο διαστημικό περιβάλλον. Στην πρώτη εκδοχή υλοποιήσαμε τον slave χρησιμοποιώντας την μέθοδο TMR σε επίπεδο ακολουθιακών κυκλωμάτων και στην δεύτερη εκδοχή χρησιμοποιήσαμε TMR και σε επίπεδο ακολουθιακών κυκλωμάτων και σε επίπεδο συνδυαστικών.

Τέλος συγκρίναμε τα αποτελέσματα των δύο αυτών υλοποιήσεων με τις αντίστοιχες υλοποιήσεις με χρήση των πρότυπων κελιών της lfoundry.

4.1 Συνοπτικά αποτελέσματα χαρακτηρισμού

Σε αυτή την ενότητα θα δούμε συνοπτικά τη σύγκριση της κάθε βιβλιοθήκης για όλα τα κελιά μαζί. Στα διαγράμματα που ακολουθούν βλέπουμε τις τιμές των χαρακτηριστικών (καθυστέρηση διάδοσης, χρόνο μετάβασης εξόδου, κατανάλωση ενέργειας και στατικής ισχύος) για όλα τα κελιά μαζί. Για τα διαγράμματα που συγκρίνουν την τροποποιημένη βιβλιοθήκη με αυτή της lfoundry, στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκης, στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκης, στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκης, στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκης, στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκη της lfoundry και στον κάθετο οι τιμές για την τροποποιημένη βιβλιοθήκης, στον οριζόντιο άξονα αντιστοιχούν οι τιμές για την βιβλιοθήκη με ακρότατο σημείο SS και στον κάθετο οι τιμές που αντιστοιχούν στην βιβλιοθήκη με ακρότατο σημείο FF. Κάθε σημείο στο διάγραμμα αντιστοιχεί στην τιμή για το ίδιο ζευγάρι τιμών (χρόνος μετάβασης εισόδου - χωρητικότητα εξόδου), με εξαίρεση την στατική ισχύ, των δύο συγκρινομενων βιβλιοθηκών και σε σχέση με τη διαγώνια γραμμή, δείχνει σε ποια βιβλιοθήκη αντιστοιχεί η μεγαλύτερη τιμή για το κάθε ζευγάρι.



Σχήμα 4.1: Γράφημα για τη σύγκριση της καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.

Στο σχήμα 4.1 βλέπουμε τη σύγκριση της βιβλιοθήκης των πρωτότυπων κελιών με την βιβλιοθήκη των τροποποιημένων κελιών για την καθυστέρηση, για ακρότατο σημείο ΤΤ. Παρατηρούμε ότι υπάρχει μικρή διαφορά μεταξύ τους, με μέση διαφορά 3,5% υπέρ της τροποποιημένης βιβλιοθήκης, όπως δείχνει το σχήμα 4.2.

Toler	ance (delay):	0.001ns and	18			<u></u>		
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	delay(ns)	8334	-0.012708	-3.51%	4.32%	-0.082085	-19.78%	6498

Σχήμα 4.2: Στατιστικά για τη σύγκριση της καθυστέρησης μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.

Στο σχήμα 4.3 βλέπουμε τη σύγκριση για της καθυστέρησης στα τροποποιημένα κελιά για ακρότατα σημεία FF και SS. Εδώ υπάρχει μεγάλη διαφορά μεταξύ τους με μέση διαφορά 50,9% υπέρ της βιβλιοθήκης για ακρότατο σημείο FF, όπως δείχνει και το σχήμα 4.4.



Σχήμα 4.3: Γράφημα για τη σύγκριση της καθυστέρησης της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

Tole	rance (delay):	0.001ns and	18					
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers (
1	delay(ns)	10600	-0.244076	-50.90% [9.37% [-1,572430	-158.91%	10600
+								

Σχήμα 4.4: Στατιστικά για τη σύγκριση της καθυστέρησης της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

Στο σχήμα 4.6 βλέπουμε τη σύγκριση της βιβλιοθήκης των πρωτότυπων κελιών με την βιβλιοθήκη των τροποποιημένων κελιών για τον χρόνο μετάβασης εξόδου, για ακρότατο σημείο ΤΤ. Παρατηρούμε ότι υπάρχει πολύ μικρή διαφορά μεταξύ τους, με μέση διαφορά 0,94% υπέρ της τροποποιημένης βιβλιοθήκης, όπως δείχνει και το σχήμα 4.5.

1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
+	trans(ns)	8334 1	-0.000266 1	-0.948 1	3 628 1	-0.176496.1	-72 68%	4963

Σχήμα 4.5: Στατιστικά για τη σύγκριση του χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.



Σχήμα 4.6: Γράφημα για τη σύγκριση του χρόνου μετάβασης εξόδου μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.

Στο σχήμα 4.7 βλέπουμε τη σύγκριση της μετάβασης εξόδου στα τροποποιημένα κελιά για ακρότατα σημεία FF και SS. Εδώ υπάρχει μεγάλη διαφορά μεταξύ τους με μέση διαφορά 47,4% υπέρ της βιβλιοθήκης για ακρότατο σημείο FF, όπως δείχνει και το σχήμα 4.8.



Σχήμα 4.7: Γράφημα για τη σύγκριση του χρόνου μετάβασης εξόδου της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

Tole	rance (trans):	0.001ns and	18					
+		*********	++		++	*******		+
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
+		+-			++			*********
1	trans(ns)	10600	-0.262868	-47.46%	11.29%	-2.539750	-61.19%	10600
+	+	+-	+		+-			+

Σχήμα 4.8: Στατιστικά για τη σύγκριση του χρόνου μετάβασης εξόδου της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

Στο σχήμα 4.9 βλέπουμε τη σύγκριση της βιβλιοθήκης των πρωτότυπων κελιών με την βιβλιοθήκη των τροποποιημένων κελιών για την κατανάλωση στατικής ισχύος ή ισχύος διαρροής, για ακρότατο σημείο ΤΤ. Παρατηρούμε ότι υπάρχει αρκετή διαφορά μεταξύ τους, με μέση διαφορά 39,5% υπέρ της βιβλιοθήκης της lfoundry, όπως δείχνει και το σχήμα 4.10.



Σχήμα 4.9: Γράφημα για τη σύγκριση της στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.

Tolerance (leakage)	: 0.001pW as	nd 1%					
Data Type	Entries	Avg Diff	Avg Diff%	Sigma 🕯 🛛	Max Diff	Max Diff%	Outliers
leakage(pW)	149	18.521538	39.50%	12.94%	40.067462	93.41%	148

Σχήμα 4.10: Γράφημα για τη σύγκριση της στατικής ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.



Σχήμα 4.11: Γράφημα για τη σύγκριση της στατικής ισχύος της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

Στο σχήμα 4.11 βλέπουμε τη σύγκριση της κατανάλωσης στατικής ισχύος ή ισχύος διαρροής στα τροποποιημένα κελιά για ακρότατα σημεία FF και SS. Εδώ υπάρχει σημαντική

διαφορά μεταξύ τους με μέση διαφορά 14,15% υπέρ της βιβλιοθήκης για ακρότατο σημείο FF, όπως δείχνει και το σχήμα 4.12.

Tol	erance (leakage): 0.001pW a	nd 1%					
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	leakage(pW)	148	-11.707745	-14.15%	4.88%	-26.242000	-27.51%	148

Σχήμα 4.12: Στατιστικά για τη σύγκριση της στατικής ισχύος της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

Στο σχήμα 4.13 βλέπουμε τη σύγκριση της βιβλιοθήκης των πρωτότυπων κελιών με την βιβλιοθήκη των τροποποιημένων κελιών για την κατανάλωση ισχύος, για ακρότατο σημείο ΤΤ. Παρατηρούμε ότι σε κάποιες περιπτώσεις υπάρχει περισσότερη κατανάλωση στην τροποποιημένη βιβλιοθήκη, αλλά στις περισσότερες στην βιβλιοθήκη της Ifoundry. Η μέση διαφορά μεταξύ τους είναι 40,3% υπέρ της βιβλιοθήκης της Ifoundry, όπως δείχνει και το σχήμα 4.14.



Σχήμα 4.13: Γράφημα για τη σύγκριση της κατανάλωσης ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.

Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
power(nJ)	11092	-0.000713 1	40.315 1	1431.735 1	0.056643 1	134434 175	7110

Σχήμα 4.14: Στατιστικά για τη σύγκριση της κατανάλωσης ισχύος μεταξύ πρωτότυπης και τροποποιημένης βιβλιοθήκης.

Στο σχήμα 4.16 βλέπουμε τη σύγκριση της στατικής ισχύος στα τροποποιημένα κελιά για ακρότατα σημεία FF και SS. Εδώ υπάρχει σημαντική διαφορά μεταξύ τους με μέση διαφορά 30,4% υπέρ της βιβλιοθήκης για ακρότατο σημείο SS, όπως δείχνει το σχήμα 4.15.

Tole	rance (power):	0.001pJ and	14	14 A	12			a 12
1	Data Type	Entries	Avg Diff	Avg Diff%	Sigma%	Max Diff	Max Diff%	Outliers
1	power(pJ)	12800	0.004631	30.43%	72.88%	0.046364	1632.51%	9076

Σχήμα 4.15: Στατιστικά για τη σύγκριση της κατανάλωσης ισχύος της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.



Σχήμα 4.16: Γράφημα για τη σύγκριση της κατανάλωσης ισχύος της τροποποιημένης βιβλιοθήκης για ακρότατα σημεία FF και SS.

4.2 Συνοπτικά αποτελέσματα της υλοποίησης του slave

Στον πίνακα 12 συνοψίζονται όλα τα αποτελέσματα των υλοποιήσεων του slave.

TMR	Υλοποίηση	Ακρότατο σημείο	Κρίσιμο μονοπάτι	Ισχύς διαρροής	Δυναμική ισχύς	Αρ. κελιών	Επιφάνεια
		SS	5,626 ns	0,017 pW	0,69 mW		
	Με τροποποιημένα κελία	FF	2,666 ns	0,014 pW	0,75 mW	371	19.392 μm²
Απλο		TT	3,506 ns	0,016 pW	0,73 mW		
4	Με πρωτότυπα κελιά	π	3,017 ns	0,011 pW	0,93 mW	370	9.726 µm²
		55	5,897 ns	0,025 pW	1,17 mW		
	Με τροποποιημένα κελία	FF	2,520 ns	0,022 pW	1,26 mW	909	30.319 µm²
τικηρες		π	4,475 ns	0,023 pW	1,22 mW		1.000000000000000000000000000000000000
. (Με πρωτότυπα κελιά	π	3,406 ns	0,016 pW	1,05 mW	912	15.204 µm²

Πίνακας 12: Συνοπτικά αποτελέσματα της υλοποίησης του slave.

4.3 Συμπεράσματα

Τα συμπεράσματα που προέκυψαν από αυτή την εργασία είναι τα εξής:

- Τα κελιά με τους δακτυλίους προστασίας καταλαμβάνουν μεγαλύτερη επιφάνεια απ' ότι τα απλά κελιά. Όχι απαραίτητα διπλάσια όπως στην περίπτωση των δικών μας κελιών. Τα κελιά μίας βιβλιοθήκης πρέπει να έχουν όλα το ίδιο ύψος, λόγω του ότι στην φυσική σχεδίαση τα κελιά τοποθετούνται το ένα δίπλα στο άλλο σε σειρές, που κάθε σειρά έχει το ύψος των κελιών. Στην περίπτωση μας το ύψος των κελιών καθορίστηκε από τα flip-flop, που έχουν το μεγαλύτερο ύψος.
- Τα αρχεία που παίρνουμε από τον χαρακτηρισμό (.lib) δεν αρκούν για να έχουμε την πλήρη εικόνα της απόδοσης ενός κυκλώματος κατά την σύνθεση. Χρειάζονται και τα αρχεία με τις πληροφορίες των μετάλλων (.lef) που περιέχουν τη γεωμετρία και τις διαστάσεις για κάθε επίπεδο μετάλλων που χρησιμοποιούνται για τις

συνδέσεις εντός των κελιών, αλλά και τις συνδέσεις των κελιών μεταξύ τους σε ολόκληρο το κύκλωμα.

 Με χρήση απλού TMR και ενός ψηφοφόρου, έχουμε χαμηλότερη ασφάλεια, σε σχέση με το πλήρες TMR, αλλά κερδίζουμε σε επιφάνεια, ταχύτητα και κατανάλωση. Με τη χρήση πλήρους TMR έχουμε μεγαλύτερη ανθεκτικότητα, αλλά έχουμε μεγάλο κόστος σε επιφάνεια και κατανάλωση. Υπάρχει διαφορά και στην ταχύτητα λόγω των μεγαλύτερων συνδυαστικών μονοπατιών, με την χρήση ψηφοφόρων στα συνδυαστικά κυκλώματα.

Με βάση το σχήμα 3.33 (β) μπορούμε να υπολογίσουμε ότι η επιφάνεια για υλοποίηση χωρίς χρήση TMR και χωρίς δακτυλίους προστασίας θα ήταν περίπου 5.500 μm². Με χρήση απλού TMR η επιφάνεια αυξάνεται σε περίπου 9.700 μm² και με χρήση πλήρους TMR η επιφάνεια αυξάνεται σε περίπου 15.000 μm². Αν θέλουμε ανθεκτικότητα και στο κλείδωμα απλού συμβάντος και προσθέσουμε δακτυλίους προστασίας, τα παραπάνω νούμερα διπλασιάζονται. Στη χειρότερη περίπτωση μιλάμε για επιφάνεια περίπου 30.000 μm², δηλαδή περίπου το εξαπλάσιο.

Σε όσο χαμηλότερο επίπεδο ενσωματώνεται η υλοποίηση του πλήρους TMR, τόσο μεγαλύτερη επιφάνεια απαιτείται και θα υπάρχουν μεγαλύτερα συνδυαστικά μονοπάτια λόγω των περισσότερων ψηφοφόρων. Δηλαδή αν η υλοποίηση του TMR ήταν σε επίπεδο υποσυστημάτων και είχαμε τρεις slave και περνάγαμε από κύκλωμα ψηφοφόρων τις εξόδους του, θα κερδίζαμε λίγο σε επιφάνεια και χρόνο, αλλά θα υπήρχαν περισσότερες πιθανότητες να υπάρξει κάποιο σφάλμα καθώς θα ήταν πιο δύσκολος ο έγκαιρος εντοπισμός του και η διόρθωση του.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος	
ASIC	ολοκληρωμένο κύκλωμα ειδικών εφαρμο-	
	γών	
Buffer	Απομονωτής	
CMOS	Συμπληρωματικός τύπος MOS	
DRAM	Δυναμική μνήμη τυχαίας προσπέλασης	
Error detection and correction	Ανίχνευση και διόρθωση σφαλμάτων	
Finite state machine	Μηχανή πεπερασμένων καταστάσεων	
Flip-flop	Δισταθής πολυδονητής	
Guard ring	Δακτύλιος προστασίας	
Hard error	Σφάλμα υλικού	
Hidden power	Κρυφή ισχύς	
Hold time	Χρόνος συγκράτησης	
Integrated circuit	Ολοκληρωμένο κύκλωμα	
Layout	Φυσική σχεδίαση	
Leakage power	Ισχύς διαρροής	
Master	Κύριος	
	Διαδικασία κατασκευής τρανζίστορ	
MOS	επίδρασης πεδίου (Μέταλλο - Οξείδιο -	
	Ημιαγωγός)	
MOSFET	Τρανζίστορ επίδρασης πεδίου MOS	
Netlist	Περιγραφή δικτυώματος	
Pin	Ακροδέκτης	
Pipeline	Σωλήνωση	
Process corner	Ακρότατο σημείο	
Recovery time	Χρόνος αποκατάστασης	
Removal time	Χρόνος απομάκρυνσης	
Reset	Επαναφορά	
Serial peripheral interface	Σειριακή διασύνδεση περιφερειακού	
Setup time	Χρόνος προετοιμασίας	
Single event burnout	Κάψιμο εξαιτίας απλού συμβάντος	
Single event effects	Φαινόμενα απλού συμβάντος	
Single event gate rupture	Ρήξη πύλης απλού συμβάντος	
Single event latch-up	Κλείδωμα απλού συμβάντος	
Single event transient	Μετάβαση απλού συμβάντος	
Single event upset	Διατάραξη απλού συμβάντος	
Slave	Υποτελής	
Slew rate	Ρυθμός ανόδου	
Soft error	Λογικό σφάλμα	
SPI	Σειριακή διεπαφή περιφερειακών	
SRAM	Στατική μνήμη τυχαίας προσπέλασης	
Standard cell	Πρότυπο κελί	
Static power	Στατική ισχύς	
Switching power	Ισχύς διακοπτικής μεταγωγής	
Template	Σχεδιότυπο	
Total ionizing dose	Ολική δόση ιονίζουσας ακτινοβολίας	
Triple mode redundancy	Λειτουργία τριπλού πλεονασμού	

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

ASIC	Application Specific Integrated Circuit
CMOS	Complementary Metal-Oxide Semiconductor
DEF	Design Exchange Format
DRAM	Dynamic Random Access Memory
ECSM	Effective Current Source Model
EDAC	Error Detection And Correction
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
IC	Integrated Circuit
LEF	Layout Exchange Format
MISO	Master In – Slave Out
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MOSI	Master Out – Slave In
NLDM	Non-Linear Delay Model
NLPM	Non-Linear Power Model
SEB	Single Event Burnout
SEE	Single Event Effects
SEGR	Single Event Gate Rupture
SEL	Single Event Latch-up
SET	Single Event Transient
SEU	Single Event Upset
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
TDI	Total Ionizing Dose
TMR	Triple modular redundancy
VLSI	Very Large-Scale Integration
KME	Κεντρική μονάδα επεξεργασίας
TEI	Τεχνολογικό Εκπαιδευτικό Ίδρυμα
ЕКПА	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

ΠΑΡΑΡΤΗΜΑ Ι

Σε αυτό το παράρτημα θα παρουσιαστεί ο κώδικας σε γλώσσα περιγραφής σεναρίων tcl, που χρησιμοποιήθηκε για τον χαρακτηρισμό των πρότυπων κελιών.

Κώδικας για την εκτέλεση του χαρακτηρισμού (char.tcl):

#####------##### Set and print user define variables ##### [pwd] ;# directory where all source data (netlist, models, etc...) are stored #set SETTINGS_FILE \${SRC_DIR}/tcl/settings.tcl set CELLS_FILE \${SRC_DIR}/template/quark \${SRC_DIR}/template/guardring_template.tcl set MODEL_INCLUDE_FILE \${SRC_DIR}/models/include_\${PROCESS} set NETLIST_DIR \${SRC_DIR}/netlist set USERDATA \${SRC_DIR}/userdata/userdata.lib set THREAD 1 :# Number of CPUs to use of set THREAD 1 ;# Number of CPUs to use on a given machine. 0: use all CPUs on machine. #####------##### Print user define settings to output #####-----puts "INFO:" puts " SRC_DIR puts " RUN_DIR = \${SRC_DIR}" = \${RUN_DIR}" = \${LIBNAME}" = \${PROCESS},\${VDD_VALUE},\${TEMP}" puts " LIBNAME puts " PVT puts TEMPLATE_FILE = \${TEMPLATE_FILE}" if { [info exists CELLS_FILE] } { puts " CELLS_FILE = \${CELLS_FILE}" } puts " MODEL_INCLUDE_FILE = \${MODEL_INCLUDE_FILE}" puts " = \${NETLIST_DIR}" NETLIST_DIR puts " USERDATA = \${USERDATA}" puts "" puts " THREAD = \${THREAD}" puts "" #####------##### Read template #####-----puts "INFO: Read template file \${TEMPLATE FILE}" source \${TEMPLATE_FILE} set cells {AND2 X0d5 AO21 X0d5 AOI21 X0d5 DFCS X1 DFC X1 DFNC X1 HA X1 INV X1 NOR2 X0d5 OAI22 X0d5 XOR2 X1 CLKBUF X1} #####------##### Read CELLS_FILE #####-----if {[info exists CELLS_FILE]} { if {[file exists \${CELLS_FILE}]} { puts "INFO: Read cell list file" source \${CELLS_FILE} } else { puts "WARNING: Specified CELLS FILE (\${CELLS FILE}) does not exist."

```
}
```

} #####------##### Set operating condition ##### puts "INFO: Set Operating Condition" set_operating_condition -voltage \${VDD_VALUE} -temp \${TEMP} #####------##### define device models #####-----puts "INFO: Define device models (spectre, define leafcell)." set var extsim model include \${MODEL INCLUDE FILE} define leafcell -type nmos -pin position {0 1 2 3} { nmos1v8hvt } define_leafcell -type pmos -pin_position {0 1 2 3} { pmos1v8hvt } #####------##### read cell netlists #####-----puts "INFO: Read cell netlist " ## setup client to read only netlist of cells being characterized set packet_cells [packet_slave_cells] if {[llength \$packet_cells]>0} { set cells \$packet_cells } ## read netlist set spicefiles {} foreach cell \$cells { lappend spicefiles \${NETLIST DIR}/\${cell}.scs } read spice -format spectre "\$MODEL INCLUDE FILE \$spicefiles" #####------##### Run characterization #####-----puts "INFO: Run Characterization" char library -auto index -extsim spectre -cells \$cells -thread \$THREAD #####------##### Write outputs #####------_____ ### Write Idb ### puts "INFO: Write Idb" if {![file exists \${RUN_DIR}/ldb]} { file mkdir \${RUN_DIR}/ldb } # In packet_arc mode, by default, existing Idb does not get overwritten. User should use -overwrite option write_ldb -overwrite \${RUN_DIR}/ldb/\${LIBNAME}.ldb ### Write Liberty ### puts "INFO: Write Libertv" if {![file exists \${RUN_DIR}/lib]} { file mkdir \${RUN_DIR}/lib } write library -user data \${USERDATA} -filename \${RUN DIR}/lib/\${LIBNAME}.lib \${LIBNAME} ### Write Datasheet ### puts "INFO: Write Datasheet" if {![file exists \${RUN_DIR}/datasheet]} { file mkdir \${RUN_DIR}/datasheet } write_datasheet -filename \${RUN_DIR}/datasheet/\${LIBNAME}_datasheet \${LIBNAME}

Κώδικας για το template (guardring_template.tcl):

set_vdd -type primary VDD \$VDD_VALUE
set_gnd -type primary VSS 0
set_gnd -no_model GND 0
set_units -leakage_power 1pw
#--------settings ------#### External Simulator (Spectre) settings ###
set_var extsim_cmd_option "+aps +spice -mt +liberate +rcopt=2"

set_var extsim_deck_header "simulator lang=spectre\nOpt1 options reltol=1e-4 \nsimulator lang=spice" set_var extsim_option "redefinedparams=ignore hier_ambiguity=lower limit=delta " set_var extsim_leakage_option "redefinedparams=ignore hier_ambiguity=lower limit=delta " ### SKI ### set_var ski_enable 1 set var ski clean mode 1 # run \$ALTOSHOME/bin/clean sm.sh to clean up inactive semaphores set_var ski_compatibility_mode 1 :# disable auto leaf cell determination set var parse auto define leafcell 0 ;# /dev/shm -use local RAM for tmp dir, /tmp - use local disk set var tmpdir /dev/shm set_var voltage_map ;# create pg_pin groups, related_power_pin / related_ground_pin 1 ;# Monitor power based on Vdd pin only set_var pin_based_power 0 set_var reset_negative_leakage_power ;# convert negative leakage current to 0 1 ;# use pre-driver waveform set_var predriver_waveform 0 set_var min_capacitance_for_outputs 1 ;# write min_capacitance attribute for output pins #----set_var slew_lower_rise 0.1 set_var slew_lower_fall 0.1 set_var slew_upper_rise 0.9 set_var slew_upper_fall 0.9 set_var measure_slew_lower_rise 0.1 set var measure slew lower fall 0.1 set_var measure_slew_upper_rise 0.9 set_var measure_slew_upper_fall 0.9 set_var delay_inp_rise 0.5 set_var delay_inp_fall 0.5 set_var delay_out_rise 0.5 set_var delay_out_fall 0.5 set_var def_arc_msg_level 0 set_var process_match_pins_to_ports 1 set var max transition 2.5e-09 set var min transition 1e-11 set_var min_output_cap 1e-15 # ----- Define templates for auto index values -----define_template -type delay \ -index_1 { 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 0.0080 0.0090 0.0100 } \ -index_2 { 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 0.0080 0.0090 0.0100 } \ delay_template_10x10 define_template -type power \ -index_1 { 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 0.0080 0.0090 0.0100 } \ -index 2 { 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 0.0080 0.0090 0.0100 } \ power_template_10x10 define_template -type delay \ -index_1 { 0.0010 0.0020 0.0030 0.0040 0.0050 } \ -index_2 { 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 0.0080 0.0090 0.0100 } delay_template_5x10 define_template -type power \ -index_1 { 0.0010 0.0020 0.0030 0.0040 0.0050 } \ -index 2 { 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 0.0080 0.0090 0.0100 } power template 5x10 define template -type constraint \ -index_1 { 0.0010 0.0020 0.0030 0.0040 0.0050 } \ -index_2 { 0.0010 0.0020 0.0030 0.0040 0.0050 } \ constraint_template_5x5 # ------ Define cells -----if {[ALAPI_active_cell "AND2_X0d5"]} { define cell \ -input { A1 A2 } \ -output { Q } \ -pinlist { A1 A2 Q } \ -delay delay_template_10x10 \ -power power_template_10x10 \ AND2 X0d5 if {[ALAPI_active_cell "NOR2_X0d5"]} { define cell \ -input { A1 A2 } \ -output { Q } \

```
-pinlist { A1 A2 Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    NOR2_X0d5
if {[ALAPI_active_cell "XOR2_X1"]} {
define_cell \
    -input { A1 A2 } \
    -output { Q } \
    -pinlist { A1 A2 Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    XOR2_X1
if {[ALAPI_active_cell "INV_X1"]} {
define_cell \
    -input { A } \
    -output { Q } \
    -pinlist { A Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    INV X1
if {[ALAPI_active_cell "CLKBUF_X1"]} {
define_cell \
    -input { A } \
    -output { Q } \
    -pinlist { A Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    ĊLKBUF X1
}
if {[ALAPI_active_cell "AO21_X0d5"]} {
define_cell \
    -input { A1 A2 B1 } \
    -output { Q } \
    -pinlist { A1 A2 B1 Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    AO21_X0d5
if {[ALAPI_active_cell "AOI21_X0d5"]} {
define_cell \
    -input { A1 A2 B1 } \
    -output { Q } \
    -pinlist { A1 A2 B1 Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    AOI21_X0d5
if {[ALAPI_active_cell "OAI22_X0d5"]} {
define_cell \
    -input { A1 A2 B1 B2 } \
    -output { Q } \
    -pinlist { A1 A2 B1 B2 Q } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    OAI22_X0d5
if {[ALAPI_active_cell "HA_X1"]} {
define_cell \
    -input { A B } \
    -output { CO SUM } \
    -pinlist { A B CO SUM } \
    -delay delay_template_10x10 \
    -power power_template_10x10 \
    HA X1
}
```

```
if {[ALAPI_active_cell "DFC_X1"]} {
define_cell \
    -clock { CP } \
    -async { CDN } \
    -input { D } \
    -output { Q QN } \
    -pinlist { CDN D CP Q QN } \
    -delay delay_template_5x10 \
    -power power_template 5x10 \
    -constraint constraint_template_5x5 \
    DFC X1
define index -pin Q -related pin CP -type {delay power} \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
     -index_2 {0.001 0.00234375 0.0046875 0.009375 0.01875 0.0375 0.075 0.15 0.3 0.45} \
    DFC_X1
define_index -pin QN -related_pin CP -type {delay power} \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
     -index_2 {0.001 0.00234375 0.0046875 0.009375 0.01875 0.0375 0.075 0.15 0.3 0.45} \
    DFC_X1
define_index -pin CP -type power \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
    DFC X1
if {[ALAPI_active_cell "DFCS_X1"]} {
define cell \
    -clock { CP } \
    -async { CDN } \
    -input { D SE SI } \
    -output { Q QN } \
    -pinlist { CDN D SE SI CP Q QN } \
    -delay delay_template_5x10 \
    -power power template 5x10 \
    -constraint constraint_template_5x5 \
    DFCS_X1
define_index -pin Q -related_pin CP -type {delay power} \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
     -index_2 {0.001 0.00234375 0.0046875 0.009375 0.01875 0.0375 0.075 0.15 0.3 0.45} \
    DFCS_X1
define_index -pin QN -related_pin CP -type {delay power} \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
     -index 2 {0.001 0.00234375 0.0046875 0.009375 0.01875 0.0375 0.075 0.15 0.3 0.45} \
    DFCS_X1
define_index -pin CP -type power \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
    DFCS_X1
if {[ALAPI_active_cell "DFNC_X1"]} {
define_cell \
    -clock { CP } \
    -async { CDN } \
    -input { D } \
    -output { Q QN } \
    -pinlist { CDN D CP Q QN } \
    -delay delay_template_5x10 \
    -power power_template_5x10 \
    -constraint constraint_template_5x5 \
    DFNC X1
define_index -pin Q -related_pin CP -type {delay power} \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
     -index_2 {0.001 0.00234375 0.0046875 0.009375 0.01875 0.0375 0.075 0.15 0.3 0.45} \
    DFNC_X1
define_index -pin QN -related_pin CP -type {delay power} \
     -index 1 {0.01 0.15625 0.3125 0.625 1.25} \
     -index_2 {0.001 0.00234375 0.0046875 0.009375 0.01875 0.0375 0.075 0.15 0.3 0.45}
    DFNC_X1
define_index -pin CP -type power \
     -index_1 {0.01 0.15625 0.3125 0.625 1.25} \
    DFNC X1}
```

ΠΑΡΑΡΤΗΜΑ ΙΙ

Σε αυτό το παράρτημα θα παρουσιάσουμε τον κώδικα σε γλώσσα περιγραφής υλικού VHDL, που χρησιμοποιήθηκε για την υλοποίηση του SPI slave.

Κώδικας για την υλοποίηση του ψηφοφόρου (Voter.vhd):

Generic Voter	
library ieee; use ieee.std_logic_1164.all; entity generic, voter is	
generic (bits: integer);	Bits for each input (Minimum 2)
Port (voter_in1: in std_logic_vector (bits-1 downto 0); voter_in2: in std_logic_vector (bits-1 downto 0 voter_in3: in std_logic_vector (bits-1 downto 0 voter_out: out std_logic_vector (bits-1 downto	 Voter's input 1); Voter's input 2); Voter's input 3 0)); Voter's out
end entity generic_voter; architecture Behavioral of generic_voter is signal temp_out: std_logic_vector (bits-1 downto 0);	
Gen_Voter: process (voter_in1, voter_in2, voter_in3) begin	
for i in bits-1 downto 0 loop temp_out(i) <= (voter_in1(i) and voter (voter_in2(i) and voter_in3(i)); end loop; end process Gen_Voter; voter_out <= temp_out; end Behavioral;	_in3(i)) or (voter_in1(i) and voter_in2(i)) or
library ieee; use ieee.std_logic_1164.all;	
entity voter is port (voter_in1: in std_logic; voter_in2: in std_logic; voter_in3: in std_logic; voter_out: out std_logic); end entity voter;	Voter's input 1 Voter's input 2 Voter's input 3 Voter's out
architecture Behavioral of voter is signal temp_out: std_logic; Attributes	
<pre> attribute keep : string; attribute keep of temp_out: signal is "TRUE"; begin temp_out <= (voter_in1 and voter_in3) or (voter_in1 a voter_out <= temp_out; end Behavioral;</pre>	and voter_in2) or (voter_in2 and voter_in3);
Κώδικας για την υλοποίηση των καταχωρητώ (Register.vhd):	ν και των καταχωρητών ολίσθησης

------ Ports -----Port (clk : in std logic; -- Clock rst : in std logic; -- Low active reset en : in std_logic; -- High active enable Din : in std_logic_vector (bits - 1 downto 0); -- Data input Dout: out std_logic_vector (bits - 1 downto 0)); -- Data output end Generic Reg: architecture Behavioral of Generic_Reg is begin process (clk, rst) begin if rst = '0' then Dout \leq (others \geq '0'); elsif clk'event and clk = clock then if en = '1' then Dout <= Din; end if; end if; end process; end Behavioral; ----- D-FF -----library ieee; use ieee.std_logic_1164.all; entity dff is generic (clock: std_logic := '1'); -- Set clock. 0: Falling, 1: Rising ----- Ports -----Port (clk : in std_logic; -- Clock rst : in std_logic; -- Low active reset en : in std_logic; -- High active enable Din : in std_logic; -- Data input Dout: out std_logic); -- Data output end dff; architecture Behavioral of dff is begin process (clk, rst) begin if rst = '0' then Dout <= '0'; elsif clk'event and clk = clock then if en = '1' then Dout \leq Din; end if; end if; end process; end Behavioral; ----- Shift Register library ieee; use ieee.std_logic_1164.all; use work.My_components.all; entity Generic_Shift_reg is generic (bits : integer := 8; -- Data width (Minimum 2) clock: std_logic := '1'); -- Set clock. 0: Falling, 1: Rising ----- Ports ------Port (clk : in std_logic; rst : in std_logic; -- Clock -- Low active reset en : in std_logic; -- High active enable reg_in : in std_logic; -- Serial input reg_out: out std_logic_vector (bits - 1 downto 0)); -- Parallel output end Generic_Shift_reg; architecture Behavioral of Generic_Shift_reg is ------ Intermediate signals ------

```
signal ff_q: std_logic_vector (bits downto 0);

begin

Shift_reg: for i in 1 to bits generate

Shift_register: dff -- Shift register unit

generic map (clock => clock)

port map (Din => ff_q(i-1),

clk => clk,

rst => rst,

en => en,

Dout => ff_q(i));

end generate Shift_reg;

ff_q(0) <= reg_in;

reg_out <= ff_q(bits downto 1);

end Behavioral;
```

Κώδικας για την υλοποίηση των καταχωρητών με χρήση TMR (TMR_unit.vhd):

1-bit TMR register				
library ieee;				
use ieee.std_logic_1164.all;				
use work.My_components.all;				
entity TMR reg is				
generic (full TMR: boolean := true;	Set TMR implementaition. Simple or full			
clock : std logic := '1');	Set clock, 0: Falling, 1: Rising			
Ports				
port (Data_in1 : in std_logic;	Data input 1			
clk1 : in std_logic;	Clock 1			
rst1 : in std_logic;	Reset 1 (active low)			
en1 : in std_logic;	Enable 1 (active high)			
Data_in2 : in std_logic;	Data input 2			
clk2 : in std_logic;	Clock 2			
rst2 : in std_logic;	Reset 2 (active low)			
en2 : in std_logic;	Enable 2 (active high)			
Data_in3 : in std_logic;	Data input 3			
clk3 : in std_logic;	Clock 3			
rst3 : in std_logic;	Reset 3 (active low)			
en3 : in std_logic;	Enable 3 (active high)			
Data_out1: out std_logic;	TMR unit out 1			
Data_out2: out std_logic;	TMR unit out 2			
Data_out3: out std_logic);	TMR unit out 3			
end entity TMR_reg;				
architecture Behavioral of TMR_reg is				
Intermediate signals				
signal reg_out: std_logic_vector (2 downto 0);	IMR Register outputs			
signal int_clk: std_logic_vector (2 downto 0);	Internal clock			
signal int_in : std_logic_vector (2 downto 0);	Internal input			
signal int_out: std_logic_vector (2 downto 0);	Internal output			
signal int_rst: std_logic_vector (2 downto 0);	Internal reset			
signal int_en : std_logic_vector (2 downto 0);	Internal enable			
begin				
Int_In <= Data_In3 & Data_In2 & Data_In1;				
$INT_CIK \leq CIK3 \& CIK2 \& CIK1;$				
$Int_Ist <= Ist_3 \& Ist_2 \& Ist_1;$				
$\operatorname{Int}_{\operatorname{en}} <= \operatorname{ens} \& \operatorname{ens} \& \operatorname{ens};$				
Data_out? <= $(0, 0)$;				
Data_out2 <= (1) , Data_out2 <= (1) ,				
$Data_0uto <- int_0ut(z),$				

Simple TMR implementation (Only regs) TMR Simple: if full TMR = false generate	
Regs: for i in 0 to 2 generate	Use 3 Dff units for TMR
Reg_units: dff	Dff units
generic map (clock => clock)	
port map (CIK => Int_CIK(I), ret => int_ret(i)	
en => int_rat();	
Din => int in(i),	
Dout => reg_out(i));	
end generate Regs,	
voter_unit: voter	Voter unit for TMR
poin map (voter_in1 => reg_out(0), voter_in2 => reg_out(1)	
voter in $3 \Rightarrow reg_out(2)$.	
voter_out => int_out(0));	
int_out(1) <= '0';	
int_out(2) <= '0'; end generate TMR_Simple;	
Full TMR implementation (All x3)	-
IMR_Full: if full_IMR = true generate	Lies 2 Off units for TMD
Regs. 101 1 11 0 to 2 generate Reg. units: dff	Ose 3 Dif units for TWR
generic map (clock => clock)	
port map (clk => int_clk(i),	
$rst => int_rst(i),$	
en => int_en(i),	
Dirit => rint_in(i), Dout => reg_out(i)):	
end generate Regs:	
Voters: for i in 0 to 2 generate	Use 3 Voters units for TMR
voter_unit: voter	Voter unit for TMR
port map (voter_in1 => reg_out(0),	
Voter_In2 => reg_out(1),	
voter_ins => reg_out(2), voter_out => int_out(i));	
end generate Voters;	
end generate TMR_Full;	
end Behavioral;	
Generic TMR register	
library ieee;	
use rece.std_logic_1164.all;	
entity Generic TMR regis	
generic (full TMR: boolean := true: Set TMR	implementation. Simple or full
bits : integer := 4; Data widt	h (Minimum 2)
clock : std_logic := '1'); Set clock	. 0: Falling, 1: Rising
port (Data_in1 : in std_logic_vector (bits-1 downto 0);	Data input 1
clk1 : in std_logic;	Clock 1
rst1 : in std_logic;	Reset 1 (active low)
enn . In siu_logic; Data in2 : in std logic vector (hits-1 downto 0):	Enable T (active high)
clk2 : in std logic:	Clock 2
rst2 : in std_logic;	Reset 2 (active low)
en2 : in std_logic;	Enable 2 (active high)
Data_in3 : in std_logic_vector (bits-1 downto 0);	Data input 3
CIK3 : IN Std_logic;	Clock 3

rst3 : in std_logic; -- Reset 3 (active low) en3 : in std logic; -- Enable 3 (active high) Data_out2: out std_logic_vector (bits-1 downto 0); Data_out2: out std_logic_vector (bits-1 downto 0); Data_out3: out std_logic_vector (bits-1 downto 0)); neric_TMR reg: -- TMR unit out 1 -- TMR unit out 2 -- TMR unit out 3 end entity Generic_TMR_reg; architecture Behavioral of Generic_TMR_reg is ------ Intermediate signals -----type vector is array (2 downto 0) of std_logic_vector (bits-1 downto 0); signal reg_out: vector; -- TMR Register outputs signal int in : vector; -- Internal input signal int_out: vector; -- Internal output -- Internal clock signal int clk: std logic vector (2 downto 0); signal int_rst: std_logic_vector (2 downto 0); -- Internal reset signal int_en : std_logic_vector (2 downto 0); -- Internal enable begin int_in(0) <= Data_in1;</pre> $int_in(1) \le Data_in2;$ $int_in(2) \le Data_in3;$ int_rst <= rst3 & rst2 & rst1;</pre> int_clk <= clk3 & clk2 & clk1; int en <= en3 & en2 & en1; Data out1 <= int out(0); Data out2 <= int out(1); Data out3 <= int out(2); ------ Simple TMR implementation (Only regs) -------TMR_Simple: if full_TMR = false generate generic map (clock => clock, bits => bits) Regs: for i in 0 to 2 generate -- Make n-bits register with TMR Reg_units: Generic_Reg -- Generic_Reg units ----rst => int_rst(i), -- $rst => int_rst(i),$ en $=> int_en(i),$ Dia $=> int_in(i)$ ---Din => int_in(i), ---Dout => reg_out(i)); end generate Regs; voter_unit: generic_voter -- generic_voter unit for TMR generic map (bits => bits) --port map (voter_in1 => reg_out(0), -voter_in2 => reg_out(1), -voter in $3 \Rightarrow reg out(2)$, -voter out = int out(0)); -int $out(1) \le (others \implies '0')$; int $out(2) \le (others \implies '0');$ end generate TMR_Simple; -- Use 3 Generic_Reg units for TMR generic map (clock => clock, --bits => bits) -port map (clk => int_clk(i), --rst => int_rst(i), --en => int_en(i), -- $Din => int_in(i),$ --Dout => reg_out(i)); --end generate Regs; Voters: for i in 0 to 2 generate -- Use 3 voters units for TMR

voter_unit: generic_voter -- generic_voter unit for TMR
generic map (bits => bits) -port map (voter_in1 => reg_out(0), -voter_in2 => reg_out(1), -voter_in3 => reg_out(2), -voter_out => int_out(i)); -end generate Voters;
end generate TMR Full;

end Behavioral;

Κώδικας για την υλοποίηση των καταχωρητών ολίσθησης με χρήση TMR (Shift_reg_TMR.vhd):

------ Shift register with TMR -----library ieee; use ieee.std_logic_1164.all; use work.My_components.all; entity Shift_reg_TMR is generic (full_TMR: boolean:= true;-- Set TMR implementation. Simple or fullbits: integer:= 8;-- Data width (Minimum 2)clock: std_logic := '1');-- Set clock. 0: Falling, 1: Rising ----- Ports ----port (reg_in1 : in std_logic; -- Serial input 1 of shift register clk1 : in std_logic; rst1 : in std_logic; en1 : in std_logic; -- Clock 1, Rising edge -- Low active reset 1 -- High active enable 1 reg_in2 : in std_logic; -- Serial input 2 of shift register clk2 : in std_logic; -- Clock 2, Rising edge rst2 : in std_logic; -- Low active reset 2 en2 : in std_logic; -- High active enable 2 -- Serial input 3 of shift register reg_in3 : in std_logic; -- Clock 3, Rising edge clk3 : in std_logic; -- Low active reset 3 rst3 : in std_logic; en3 : in std_logic; -- High active enable 3 reg_out1: out std_logic_vector (bits-1 downto 0); -- Parallel output of shift register reg out2: out std logic vector (bits-1 downto 0); -- Parallel output of shift register reg_out3: out std_logic_vector (bits-1 downto 0)); -- Parallel output of shift register end Shift reg TMR; architecture rtl of Shift_reg_TMR is ------ Intermediate signals -----type vector is array (2 downto 0) of std logic vector (bits-1 downto 0); signal int out: vector; -- Internal output signal reg_out: vector; -- Internal outputs of shift register signal int in : std logic vector (2 downto 0); -- Internal input -- Internal input -- Internal clock signal int_clk: std_logic_vector (2 downto 0); signal int_rst: std_logic_vector (2 downto 0); -- Internal reset signal int_en : std_logic_vector (2 downto 0); -- Internal enable begin int_rst <= rst3 & rst2 & rst1; int_clk <= clk3 & clk2 & clk1; int_en <= en3 & en2 & en1; int_in <= reg_in3 & reg_in2 & reg_in1; reg_out1 <= int_out(0); reg_out2 <= $int_out(1)$; reg out3 $\leq int_out(2);$ Shift regs: for i in 2 downto 0 generate Shift register: Generic Shift reg -- Shift register unit generic map (bits => bits,

clock => clock)	Rising edge
port map (reg_in => int_in(i),	
clk => int_clk(i),	
$rst => int_rst(i),$	
en => int_en(i),	
reg_out => reg_out(i));	
end generate Shift_regs;	
Simple TMR implementation (Only regs)	
TMR_Simple: if full_TMR = false generate	
Voter: generic_voter	Voter unit
generic map (bits => bits)	
port map (voter_in1 => reg_out(0),	
voter_in2 => reg_out(1),	
voter_in3 => reg_out(2),	
voter_out => int_out(0));	
int_out(1) <= (others => '0');	
int_out(2) <= (others => '0');	
end generate TMR_Simple;	
Full TMR implementation (All x3)	
TMR_Full: if full_TMR = true generate	
Voters: for i in 2 downto 0 generate	3x Voters
Voter: generic_voter	Voter unit
generic map (bits => bits)	
port map (voter_in1 => reg_out(0),	
voter_in2 => reg_out(1	l),
voter_in3 => reg_out(2	2),
voter_out => int_out(i));
end generate Voters;	
end generate TMR_Full;	

end rtl;

Κώδικας για την υλοποίηση των μετρητών με χρήση TMR (Counter_TMR.vhd):

library ieee;			
use ieee.std_logic_1164	4.all;		
use ieee.std_logic_arith	.all;		
use ieee.std_logic_unsi	gned.all;		
use work.My_component	nts.all;		
entity Counter_TMR is			
generic (full_TMF	R: boolean := true;	Set TMR	implementaition. Simple or full
bits	: integer := 4;	Bits of co	unter (Minimum 2)
cloc	k : std_logic := '1');	Set clock.	0: Falling, 1: Rising
	Ports		
port (clk1	: in std_logic;		Clock 1
en1	: in std_logic;		Active high enable 1
rst1	: in std_logic;		Active low reset 1
clk2	: in std_logic;		Clock 2
en2	: in std_logic;		Active high enable 2
rst2	: in std_logic;		Active low reset 2
clk3	: in std_logic;		Clock 3
en3	: in std_logic;		Active high enable 3
rst3	: in std_logic;		Active low reset 3
counter_c	out1: out std_logic_vector (b	its-1 downto 0);	Counter output
counter_c	out2: out std_logic_vector (b	its-1 downto 0);	Counter output
counter_c	out3: out std_logic_vector (b	its-1 downto 0));	Counter output
end entity Counter_TMF	र;		
architecture rtl of Count	er_TMR is		
	Intermediate signals		

type vector is array (2 downto 0) of std_logic_vector (bits-1 downto 0); signal voter out : vector; -- Voter out signal cnt out : vector; -- Adder out _logic_vector (2 downto 0); -- Enable reg in : std_logic_vector (2 downto 0); -- Enable reg out signal en_in : std_logic_vector (2 downto 0); signal en begin en_in <= en3 & en2 & en1; counter out1 <= voter out(0); counter_out2 <= voter_out(1); counter_out3 <= voter_out(2); ------ Simple TMR implementation (Only regs) -------TMR_Simple: if full_TMR = false generate

 units: Generic_TMR_reg
 -- Generic_TMR_Reg units

 generic map (full_TMR => full_TMR, bits => bits, clock => clock)
 -- Generic_TMR_Reg units

 -- Set TMR implementation. Simple or full

 -- Data width (Minimum 2)

 -- Set clock. 0: Falling, 1: Rising

 Reg_units: Generic_TMR_reg ----- Ports ----port map (Data_in1 => $cnt_out(0)$, -- $Data_in2 => cnt_out(0),$ --- $Data_in3 => cnt_out(0),$ -clk1 => clk1, -clk2 => clk2, -clk3 => clk3, --rst1 => rst1. --rst2 => rst2.=> rst3, rst3 --=> en(0), en1 --=> en(0), en2 --=> en(0), en3 --Data out1 => voter out(0), --Data out2 => voter out(1), ---Data_out3 => voter_out(2)); $cnt_out(2) \le (others \implies '0');$ $cnt_out(1) \le (others \implies '0');$ $cnt_out(0) \le voter_out(0) + 1;$ -- Adder end generate TMR_Simple; ----- Full TMR implementation (All x3) -----units: Generic_TMR_reg --- Generic_TMR_Reg units generic map (full_TMR => full_TMR, bits => bits, clock => clock) --- Set clock. 0: Falling, 1: Rising port map (Data in1 => cont_crit(0) TMR_Full: if full_TMR = true generate Reg_units: Generic_TMR_reg port map (Data_in1 => cnt_out(0), Data in $2 \Rightarrow cnt out(1)$, -- $Data_in3 => cnt_out(2),$ -clk1 => clk1, -clk2 => clk2, -clk3 => clk3, --=> rst1, rst1 --=> rst2, rst2 -rst3 => rst3, -en1 => en(0),--en2 => en(1), -en3 => en(2),--Data_out1 => voter_out(0), ---Data_out2 => voter_out(1), ---Data_out3 => voter_out(2)); ---
Counters: for i in 0 to 2 generate cnt_out(i) <= voter_out(i) + 1; end generate Counters; end generate TMR_Full;	Make 3 copies of counter Adder	
cnt en rea: TMR rea		
generic map (full_TMR => full_TMR,	Set TMR implementaition	
clock => '1')	Rising edge	
port map (Data in1 => en in(0),		
Data_in2 \Rightarrow en_in(1),		
$Data_in3 => en_in(2),$		
clk2 => clk2,		
clk3 => clk3,		
rst1 => rst1,		
$rst_2 => rst_2$, $rst_3 => rst_3$.		
en1 => '1',	Always enabled	
en2 => '1',	Always enabled	
en3 => 1', Data out1 => en(0)	Always enabled	
$Data_out2 => en(0),$ Data_out2 => en(1),		
Data_out3 => en(2));		
end rtl;		
library ieee; use ieee.std_logic_1164.all; use ieee.std_logic_arith.all; use ieee.std_logic_unsigned.all;	ρους της κ.ιν.ε (FSM_comb.vna):	
entity FSM_comb is		
port (MOSI : in std_logic;	Serial input from master	
count_data : in std_logic_vector (3 downto 0); count_addr : in std_logic_vector (1 downto 0):	Counter for address	
data_in : in std_logic_vector (7 downto 0);	Data in from reg file (via register)	
CurrState : in std_logic_vector (3 downto 0);	Out of state register (taking addr mode)	
CurrState_wr. In std_logic_vector (1 downto 0); CurrState_rd: in std_logic_vector (1 downto 0);	Output of state register (whiling mode)	
wr_en : out std_logic; Wri	ite enable signal for reg file (via register)	
en_4bit : out std_logic;	Enable for address shift register	
input 4bit : out std_logic;	Data input for address shift register	
input_8bit : out std_logic;	Data input for data shift register	
en_data_cnt : out std_logic;	Enable for data counter	
en_addr_cnt : out std_logic; MISO : out std_logic:	Enable for address counter	
NextState : out std_logic_vector (3 downto 0);	Input of state reg (taking addr mode)	
NextState_wr: out std_logic_vector (1 downto 0);	Input of state reg (writing mode)	
NextState_rd: out std_logic_vector (1 downto 0));	Input of state reg (reading mode)	
architecture Behavioral of FSM_comb is		
Constants for FSM encoding		
constant IDLE : std_logic_vector (3 downto 0)	:= "0000"; "0011":	
constant MODE : std logic vector (3 downto 0)	:= "1100";	
constant WAITING : std_logic_vector (3 downto 0)	:= "1111";	
constant wr_IDLE : std_logic_vector (1 downto 0) := "01 constant STORE_DATA : std_logic_vector (1 downto 0)	"; := "10";	

```
constant rd_IDLE : std_logic_vector (1 downto 0) := "01";
      constant READ DATA
                             : std logic vector (1 downto 0) := "10";
      ------ Intermediate signals ------
      signal move_on_wr : std_logic;
                                               -- Writing mode
      signal move on rd : std logic;
                                               -- Reading mode
      signal MISO1
                       : std_logic;
                                               -- MISO for write fsm
      signal MISO2
                         : std_logic;
                                                -- MiSO for read fsm
begin
      MISO <= MISO1 or MISO2;
read_address: process (CurrState, MOSI, data_in, count_addr)
      begin
            case CurrState is
                  when IDLE =>
                        en_addr_cnt <= '1';
                        en_data_cnt <= '0';
                        MISO1 <= '0';
                        move_on_wr <= '0';
                        move_on_rd <= '0';
                        en_4bit <= '1';
                        en_8bit <= '0';
                        input_4bit <= MOSI;
                        NextState <= TAKE ADDR;
                  when TAKE ADDR =>
                        en data cnt <= '0';
                        MISO1 <= '0':
                        move_on_wr <= '0';
                        move_on_rd <= '0';
                        en 4bit <= '1';
                        en 8bit <= '0';
                        input_4bit <= MOSI;
                        if count addr < 3 then
                               en addr cnt <= '1';
                               NextState <= TAKE_ADDR;</pre>
                        else
                               en_addr_cnt <= '0';
                               NextState <= MODE;
                        end if:
                  when MODE =>
                        en_addr_cnt <= '0';
                        en_data_cnt <= '1';
                        en_4bit <= '0';
                        input_4bit <= '0';
                        NextState <= WAITING;</pre>
                        if MOSI = '0' then
                                                 -- Write data from master and send them to reg file
                               MISO1 <= '0':
                               move on wr \leq 1';
                               move_on_rd <= '0';
                               en 8bit <= '1';
                        else
                                                 -- Read data from reg file and send them to master
                               MISO1 \le data_in(7);
                                                       -- Send 1st bit, if slave its on reading mode
                               move_on_wr <= '0';
                               move_on_rd <= '1';
                               en_8bit <= '0';
                        end if:
                  when WAITING =>
                        en_addr_cnt <= '0';
                        en_data_cnt <= '1';
                        MISO1 <= '0';
                        move_on_wr <= '0';
                        move_on_rd <= '0';
                        en_4bit <= '0';
```

```
en_8bit <= '1';
                  input 4bit <='0';
                  NextState <= WAITING;
            when others =>
                  en_addr_cnt <= '0':
                  en data cnt <= '0';
                  MISO1 <= '0';
                  move on wr \le 0';
                  move_on_rd <= '0';
                  en_4bit <= '0';
                  en 8bit <= '0';
                  input 4bit <= MOSI;
                  NextState <= IDLE;
      end case:
end process read_address;
write_data_fsm: process (CurrState_wr, MOSI, count_data, move_on_wr)
begin
      case CurrState_wr is
            when wr_IDLE =>
                  wr_en <= '0';
                  input_8bit <= MOSI;
                  if move on wr = '0' then
                        NextState_wr <= wr_IDLE;</pre>
                  else
                        NextState wr <= STORE DATA;
                  end if:
            when STORE DATA =>
                  input 8bit <= MOSI;
                  if count data < 8 then
                        wr_en <= '0';
                        NextState wr <= STORE DATA;
                  elsif count data = 8 then
                         wr en <= '1';
                         NextState_wr <= STORE_DATA;
                  else
                         wr_en <= '0';
                         NextState_wr <= wr_IDLE;</pre>
                  end if;
            when others =>
                  input_8bit <= '0';
                  wr_en <= '0';
                  NextState_wr <= wr_IDLE;</pre>
      end case;
end process write data fsm;
read_data_fsm: process (CurrState_rd, count_data, data_in, move_on_rd)
begin
      case CurrState_rd is
            when rd IDLE =>
                  if move on rd = '1' then
                         MISO2 <= data_in (7 - conv_integer(count_data));
                         NextState_rd <= READ_DATA;</pre>
                  else
                         MISO2 <= '0':
                         NextState_rd <= rd_IDLE;</pre>
                  end if:
            when READ_DATA =>
                  if count_data <= 7 then
                        MISO2 <= data_in (7 - conv_integer(count_data));
                         NextState_rd <= READ_DATA;</pre>
                  else
                         MISO2 <= '0';
```

NextState_rd <= rd_IDLE; end if; when others => MISO2 <= '0'; NextState_rd <= rd_IDLE; end case; end process read_data_fsm;

end Behavioral;

Κώδικας για την υλοποίηση με χρήση απλού TMR του ανώτερου επιπέδου του slave (SPI_slave_TMR.vhd):

library ieee; use ieee.std logic 1164.all; use ieee.std logic arith.all; use ieee.std logic unsigned.all; use work.My_components.all; entity spi_slave_TMR is port (i_SCLK : in std_logic; -- Clock : in std_logic; -- Reset i_rstn : in std_logic; -- Slave enable (low active) i CSN : in std_logic; -- Serial input from master i MOSI i_data_in : in std_logic_vector (7 downto 0); -- Data in from reg file for read mode o_wr_en : out std_logic; o_MISO : out std_logic; -- Write enable -- Serial output to master o_addr_out: out std_logic_vector (3 downto 0); -- Address for writing data to reg file o_data_out: out std_logic_vector (7 downto 0)); -- Data out to reg file end entity spi_slave_TMR; architecture Stractural of spi_slave_TMR is ----- Intermediate signals ------: std logic vector (1 downto 0); -- Output of state register (writing mode) signal CurrState wr : std_logic_vector (1 downto 0); -- Input of state register (writing mode) signal NextState_wr signal CurrState rd : std logic vector (1 downto 0); -- Output of state register (reading mode) signal NextState rd: std logic vector (1 downto 0); -- Input of state register (reading mode) signal CurrState : std logic vector (3 downto 0); -- Output of state register (taking address mode) signal NextState : std logic vector (3 downto 0); -- Input of state register (taking address mode) signal data in : std logic vector (7 downto 0); -- Data in. For reading mode signal count_data : std_logic_vector (3 downto 0); -- Counter's out for data signal count_addr : std_logic_vector (1 downto 0); -- Counter's out for address signal wr_en_d : std_logic; -- Input for write enable register signal wr_en_q : std_logic; -- Output for write enable register : std_logic; signal en 4bit -- Enable for data shift register signal en_8bit : std_logic; -- Enable for address shift register signal en_data_cnt: std_logic; -- Input for counter_enable register (data counter enable) signal en addr cnt: std logic; -- Input for counter enable register (address counter enable) -- Shift register's inpput (Address) signal input_4bit : std_logic; signal input_8bit : std_logic; -- Shift register's input (Data) signal int rst -- Internal reset (low active). : std logic; signal MISO : std_logic; -- Serial output to master --signal MOSI : std_logic; -- Serial input from master begin int_rst <= i_rstn and (not i_CSN); -- Reset (active low) when slave is inactive and on system reset o wr en $\leq wr en q;$ addr shift reg: Shift reg TMR -- Shift register for address generic map (full TMR => false, -- Set simple TMR implementation. -- Bits of Shift register (Minimum 2) bits => 4. clock => '1')-- Rising edge ----- Ports ------=> i_SCLK, port map (clk1

clk2 => i_SCLK, clk3 => i SCLK, rst1 => int rst,rst2 => int_rst, rst3 => int rst, en1 => en_4bit, en2 => en_4bit, en3 => en 4bit, reg_in1 => input_4bit, reg_in2 => input_4bit, reg in $3 \Rightarrow$ input 4bit, $reg_out1 => o_addr_out);$ data_shift_reg: Shift_reg_TMR -- Shift register for daata -- Set simple TMR implementation. generic map (full_TMR => false, -- Bits of Shift register (Minimum 2) bits => 8, clock => '1') -- Rising edge ----- Ports ----port map (clk1 => i_SCLK, $clk2 => i_SCLK,$ clk3 => i_SCLK, rst1 => int_rst, rst2 => int rst, rst3 => int rst, en1 => en 8bit, en2 => en 8bit. en3 => en 8bit, reg_in1 => input_8bit, reg_in2 => input_8bit, reg_in3 => input_8bit, $reg_out1 => o_data_out);$ data_counter: Counter_TMR -- Counter for write mode MR => false, bits => 4, clock => '0') generic map (full_TMR => false, -- Set simple TMR implementation. -- Bits of counter (Minimum 2) -- Falling edge ----- Ports -----port map (clk1 => i_SCLK, $clk2 => i_SCLK,$ clk3 => i_SCLK, $rst1 => int_rst,$ rst2 => int_rst, rst3 => int_rst, en1 => en_data_cnt, en2 => en_data_cnt, en3 => en data cnt,counter_out1 => count_data); address counter: Counter TMR r_TMR-- Counter for read modeMR => false,-- Set simple TMR implementaition.bits => 2,-- Bits of counter (Minimum 2)clock => '1')-- Rising edge -- Counter for read mode generic map (full_TMR => false, bits => 2, ----- Ports -----port map (clk1 => i_SCLK, clk2 => i_SCLK, clk3 => i_SCLK, $rst1 => int_rst,$ rst2 => int_rst, rst3 => int_rst, en1 => en_addr_cnt, en2 => en_addr_cnt, en3 => en_addr_cnt, counter_out1 => count_addr); wr_en_reg: TMR_reg -- Register for write enable

generic map (full_TMR => false, -- Set Full TMR implementaition clock => '1') -- Rising edge ----- Ports -------- Poits ta_in1 => wr_en_d, Data_in2 => wr_en_d, Data_in3 => wr_en_d, '''4 => i_SCLK, port map (Data_in1 => wr_en_d, --------------clk3 => i_SCLK, --rst1 => int_rst, --rst1 => int_rst, rst2 => int_rst, rst3 => int_rst, en1 => '1', en2 => '1', en3 => '1', Data_out1 => wr_en_q); -------- Always enabled -- Always enabled -- Always enabled --data_in_reg: Generic_TMR_reg -- Register for data_in generic map (full_TMR => false, -- Set Full TMR implementaition bits => 8, -- Data width clock => '0') -- Falling edge ----- Ports ----port map (Data_in1 => i_data_in, Data_in2 => i_data_in, Data_in3 => i_data_in, clk1 => i_SCLK, clk2 => i_SCLK, clk3 => i_SCLK, rst1 => int_rst, rst2 => int_rst, rst3 => int_rst, en1 => '1', en2 => '1', Data_out1 => data_in); 0_reg: TMR_reg generic map (full_TMR => false, clock => '0') --------- Ports -------port map (Data_in1 => MISO, port map (Data_in1 => i_data_in, ---------------------------- Always enabled -- Always enabled -- Always enabled MISO_reg: TMR_reg -- Register for MISO output -- Set Full TMR implementaition -- Falling edge port map (Data_in1 => MISO, -- Always enabled -- Always enabled -- Always enabled Read_state_reg: Generic_TMR_reg -- State register for reading mode FSM generic map (full_TMR => false, -- Set Full TMR implementaition -- Data width -- Rising edge ----- Ports ----port map (Data_in1 => NextState_rd, ---Data_in2 => NextState_rd, --Data_in2 => NextState_rd, Data_in3 => NextState_rd, --clk1 => i_SCLK, ---

clk2 $=> i_SCLK$, clk3 => i_SCLK, ---=> int_rst, --rst1 rst2 => int_rst, --------- Always enabled -- Always enabled -- Always enabled rst3 => int rst, => '1', en1 => '1'. en2 => '1', -- Always enabled en3 Data_out1 => CurrState_rd); -- State register for writing mode FSM Write_state_reg: Generic_TMR_reg -- Set Full TMR implementaition generic map (full_TMR => false, bits => 2, -- Data width clock => '0') -- Falling edge ----- Ports ----port map (Data_in1 => NextState_wr, --Data_in2 => NextState_wr, Data_in3 => NextState_wr, ----- $clk1 => i_SCLK,$ -- $clk2 => i_SCLK,$ --clk3 => i_SCLK, --rst1 => int_rst, --- $\begin{array}{rcl} rst1 & => int_rst, & -- \\ rst2 & => int_rst, & -- \\ en1 & => '1', & -- \\ en2 & => '1', & -- \\ en3 & => '1', & -- \\ Data_out1 => CurrState_wr); & -- \\ rst2 & Conoria TMP req & -- \\ State register for \\ req & -- \\ req$ take_addr_state_reg: Generic_TMR_reg -- State register for writing mode FSM -- Set Full TMR implementation generic map (full_TMR => false, bits =>4, -- Data width clock => '0')-- Falling edge ----- Ports ----port map (Data_in1 => NextState, ---Data_in2 => NextState, ---Data_in3 => NextState, --clk1 => i_SCLK, --clk2 => i_SCLK, -- $clk3 => i_SCLK,$ --------- Always enabled -- Always enabled -rst1 => int_rst, --rst2 => int_rst, rst3 => int_rst, en1 => '1', en2 => '1', => '1', en3 Data_out1 => CurrState); FSM: FSM comb --port map (MOSI => i_MOSI, count_data => count_data, --count_addr => count_addr, ---=> data in, -data in CurrState => CurrState, --CurrState wr => CurrState wr, ---CurrState_rd => CurrState_rd, --=> wr_en_d, wr_en -en_4bit => en_4bit, -en_8bit => en_8bit, --input_4bit => input_4bit, input_8bit => input_8bit, en_data_cnt => en_data_cnt, en_addr_cnt => en_addr_cnt, -----------MISO => MISO, ---

NextState => NextState,	
NextState_wr => NextState_wr,	
NextState_rd => NextState_rd);	

end architecture Stractural;

Κώδικας για την υλοποίηση με χρήση πλήρους TMR του ανώτερου επιπέδου του slave (SPI slave full TMR.vhd):

library ieee; use ieee.std logic 1164.all; use work.My_components.all; entity spi_slave_full_TMR is -- Clock 1 port (i SCLK1 : in std logic; -- Slave enable (low active) 1 i CSN1 : in std logic; i rstn1 : in std logic; -- Reset 1 : in std logic; -- Serial input from master 1 i MOSI1 -- Data in from reg file for read mode 1 i_data_in1 : in std_logic_vector (7 downto 0); i_SCLK2 : in std_logic; -- Clock 2 i CSN2 : in std_logic; -- Slave enable (low active) 2 : in std_logic; -- Reset 2 i_rstn2 i MOSI2 : in std logic; -- Serial input from master 2 i_data_in2 : in std_logic_vector (7 downto 0); -- Data in from reg file for read mode 2 i_SCLK3 : in std_logic; -- Clock 3 -- Slave enable (low active) 3 i_CSN3 : in std_logic; i rstn3 : in std logic; -- Reset 3 i MOSI3 : in std_logic; -- Serial input from master 3 i_data_in3 : in std_logic_vector (7 downto 0); -- Data in from reg file for read mode 3 o_MISO1 : out std_logic; -- Serial output to master 1 o_wr_en1 : out std_logic; -- Write enable 1 o_addr_out1: out std_logic_vector (3 downto 0); -- Address for writing data to reg file 1 o_data_out1: out std_logic_vector (7 downto 0); -- Data out to reg file 1 o_MISO2 : out std_logic; -- Serial output to master 2 o wr en2 : out std logic; -- Write enable 2 o addr out2: out std logic vector (3 downto 0); -- Address for writing data to reg file 2 o data out2: out std logic vector (7 downto 0); -- Data out to reg file 2 o MISO3 : out std logic; -- Serial output to master 3 o wr en3 : out std logic; -- Write enable 3 o_addr_out3: out std_logic_vector (3 downto 0); -- Address for writing data to reg file 3 o_data_out3: out std_logic_vector (7 downto 0)); -- Data out to reg file 3 end spi_slave_full_TMR; architecture Stractural of spi_slave_full_TMR is ------ Intermediate signals -----type vector1 is array (2 downto 0) of std_logic_vector (1 downto 0); type vector2 is array (2 downto 0) of std_logic_vector (3 downto 0); -type vector3 is array (2 downto 0) of std_logic_vector (7 downto 0); signal NextState_rd -- Input of state register (reading mode) : vector1; signal CurrState_rd -- Output of state register (reading mode) : vector1; signal CurrState wr : vector1; -- Output of state register (writing mode) signal NextState wr : vector1; -- Input of state register (writing mode) signal count_addr : vector1; -- Counter's out for address signal count data : vector2; -- Counter's out for data signal CurrState : vector2; -- Output of state register (taking address mode) -- Input of state register (taking address mode) signal NextState : vector2; signal data in d : vector3; -- Data in from data in reg. For read mode signal data_in_q : vector3; -- Data in from data in reg. For read mode signal clk : std logic vector (2 downto 0); -- Internal clock signal MOSI : std logic vector (2 downto 0); -- Serial input from master : std_logic_vector (2 downto 0); -- Output of write enable register signal wr_en_q signal wr_en_d : std_logic_vector (2 downto 0); -- Input for write enable register signal en_4bit_d : std_logic_vector (2 downto 0); -- Enable for data shift register. Voter in

signal en_4bit_q : std_logic_vector (2 downto 0); -- Enable for data shift register. Voter out signal en 8bit d : std logic vector (2 downto 0); -- Enable for address shift register. Voter in signal en_8bit_q : std_logic_vector (2 downto 0); -- Enable for address shift register. Voter out signal input_4bit : std_logic_vector (2 downto 0); -- Address shift register's input signal input 8bit : std logic vector (2 downto 0); -- Data shift register's input signal en data cnt d: std logic vector (2 downto 0); -- In for couun en reg (data counter enable) signal en_data_cnt_q: std_logic_vector (2 downto 0); -- Out for count en reg (data counter enable) signal en_addr_cnt_d: std_logic_vector (2 downto 0); -- In for count en reg (address counter enable) signal en_addr_cnt_q: std_logic_vector (2 downto 0); -- Out for count en reg (addr counter enable) signal MISO_d : std_logic_vector (2 downto 0); -- Input of MISO voter -- Internal reset : std logic vector (2 downto 0); signal int rst begin <= i_SCLK3 & i_SCLK2 & i_SCLK1; clk MOSI <= i_MOSI3 & i_MOSI2 & i_MOSI1; o_wr_en1 <= wr_en_q(0); o_wr_en2 <= wr_en_q(1); o_wr_en3 <= wr_en_q(2); $data_in_d(0) \le i_data_in1;$ data_in_d(1) <= i_data_in2; data_in_d(2) <= i_data_in3; int_rst(0) <= i_rstn1 and (not i_CSN1); -- Reset (active low) when slave is inactive or reset 1 int rst(1) <= i rstn2 and (not i CSN2); -- Reset (active low) when slave is inactive or reset 2</pre> int_rst(2) <= i rstn3 and (not i CSN3); -- Reset (active low) when slave is inactive or reset 3</pre> shift_reg: Shift_reg_TMR --- Shift register for address generic map (full_TMR => true, bits => 4, clock => '1') -- Rising edge addr shift reg: Shift reg TMR ----- Ports ----port map (reg_in1 => input_4bit(0), clk1 => clk(0),-rst1 \Rightarrow int rst(0), -en1 $=> en_4bit_q(0),$ - $reg_in2 => input_4bit(1),$ -clk2 => clk(1),-rst2 => int_rst(1), -- $en2 => en_4bit_q(1),$ -- $reg_in3 => input_4bit(2),$ --clk3 => clk(2),-- $rst3 => int_rst(2),$ -- $en3 => en_4bit_q(2),$ -- $reg_out1 => o_addr_out1,$ --reg out2 => o addr out2, --reg out3 => o addr out3); --data shift reg: Shift reg TMR -- Shift register for data generic map (full TMR = true, -- Set Full TMR implementaition bits => 8, -- Bits of Shift register (Minimum 2) clock => '1')-- Rising edge ----- Ports ----port map (reg_in1 => input_8bit(0), --clk1 => clk(0),--rst1 \Rightarrow int rst(0), --- \Rightarrow en 8bit q(0), en1 -reg_in2 => input_8bit(1), -clk2 => clk(1),- $rst2 => int_rst(1),$ --en2 => $en_8bit_q(1)$, --reg_in3 => input_8bit(2), --clk3 => clk(2),-- $rst3 => int_rst(2),$ --en3 $=> en_8bit_q(2),$ ---

 $reg_out1 => o_data_out1,$ reg out2 => o data out2, ----reg_out3 => o_data_out3); data_counter: Counter_TMR -- Counter for data -- Set Full TMR implementaition => clk(0),port map (clk1 --------------------------counter_out1 => count_data(0), - $counter_out2 => count_data(1),$ --counter_out3 => count_data(2)); -address_counter: Counter_TMR -- Counter for address generic map (full_TMR => true, -- Set Full TMR implementaition bits => 2, -- Bits of counter (Minimum 2) clock => '1'-- Rising edge ----- Ports -----2 => clk(1), 2 => clk(2), ret(port map (clk1 => clk(0), --clk2 ---=> clk(2), => int_rst(0), :¬t_rst(1), --clk3 --rst1 => int_rst(1), => int_rst(2), --=> en_addr_cnt_q(0), --=> en_addr_cnt_q(1), --=> en_addr_cnt_q(2), --1 -> count_addr(0), -rst2 rst3 en1 en2 en3 counter_out1 => count_addr(0), counter_out2 => count_addr(1), --counter_out3 => count_addr(2)); --wr_en_reg: TMR_reg generic map (full_TMR => true, -- Set Full TMR implementaition MR => true,clock => '1')-- Rising edge ----- Ports ----port map (Data_in1 => wr_en_d(0), ---Data in $2 \Rightarrow wr en d(1)$, --Data_in3 => wr_en_d(2), --clk1 => clk(0),--clk2 => clk(1),-clk3 => clk(2),---=> int_rst(0), --rst1 --rst2 => int_rst(1), \Rightarrow int_rst(2), rst3 en1 => '1', -- Always enabled => '1', en2 -- Always enabled => '1', en3 -- Always enabled $Data_out1 => wr_en_q(0),$ --- $Data_out2 => wr_en_q(1),$ --- $Data_out3 => wr_en_q(2));$ -data_in_reg: Generic_TMR_reg -- Register for data_in generic map (full_TMR => true, -- Set Full TMR implementaition -- Data width bits => 8, clock => '0')-- Falling edge ----- Ports ------

port map (Data_in1 => data_in_d(0), $Data_in2 => data_in_d(1),$ --Data_in3 => data_in_d(2), --=> clk(0),clk1 --=> clk(1), clk2 --=> clk(2), clk3 -rst1 => int_rst(0), ---= int rst(1), --rst2 \Rightarrow int_rst(2), --rst3 => '1', -- Always enabled en1 => '1', en2 -- Always enabled => '1', en3 -- Always enabled $Data_out1 => data_in_q(0),$ -- $Data_out2 => data_in_q(1),$ --Data_out3 => data_in_q(2)); ---MISO_reg: TMR_reg generic map (full_TMR => true, -- Set Full TMR implementaition clock => '0') -- Falling edge ----- Ports ----port map (Data_in1 => MISO_d(0), --- $Data_in2 => MISO_d(1),$ -- $Data_in3 => MISO_d(2),$ --clk1 => clk(0),--clk2 => clk(1),--clk3 => clk(2),---=> int_rst(0), --rst1 --rst2 = int_rst(1), --rst3 = int_rst(2), => '1', -- Always enabled en1 => '1', en2 -- Always enabled => '1', en3 -- Always enabled Data out1 => o MISO1, $Data_out2 => o_MISO2,$ --Data_out3 => o_MISO3); ---Read_state_reg: Generic_TMR_reg -- State register for reading mode FSM generic map (full_TMR => true, -- Set Full TMR implementaition -- Data width bits => 2, clock => '1') -- Rising edge ----- Ports ----port map (Data_in1 => NextState_rd(0), ---Data_in2 => NextState_rd(1), ---Data_in3 => NextState_rd(2), -clk1 => clk(0),-clk2 => clk(1),-clk3 => clk(2),-- \Rightarrow int_rst(0), --rst1 --rst2 = int_rst(1), rst3 => int_rst(2), ---=> '1', en1 -- Always enabled => '1', en2 -- Always enabled => '1', en3 -- Always enabled Data out1 => CurrState rd(0), --Data out2 => CurrState_rd(1), --Data_out3 => CurrState_rd(2)); Write state_reg: Generic_TMR_reg -- State register for writing mode FSM generic map (full_TMR => true, -- Set Full TMR implementaition bits => 2, -- Data width clock => '0') -- Falling edge ----- Ports ----port map (Data_in1 => NextState_wr(0), ---Data_in2 => NextState_wr(1), ---

$\begin{array}{llllllllllllllllllllllllllllllllllll$	
Ports	
port map (Data_In1 => NextState(0),	
Data in $3 => NextState(2)$.	
clk1 => clk(0),	
clk2 => clk(1),	
clk3 => clk(2),	
$rst1 => int_rst(0),$ ret2 => int_ret(1)	
$rst_3 => int_rst_2(2)$	
en1 => '1',	Always enabled
en2 => '1',	Always enabled
en3 => '1',	Always enabled
Data_out1 => CurrState(0),	
Data_out2 => CurrState(1), Data_out3 => CurrState(2));	
ESMs: for i in 2 downto 0 generate	3x FSMs
FSM: FSM_comb	
port map (MOSI => MOSI(i),	
count_data => count_data(i)),
count_addr => count_addr(i)),
uala_III => uala_III_q(I), CurrState -> CurrState(i)	
CurrState wr => CurrState w	/r(i)
CurrState_rd => CurrState_rd	d(i),
wr_en => wr_en	_d(i),
en_4bit => en_4bit_d(i),	
en_80It => en_80It_0(I), input 4bit => input 4bit(i)	
input_4bit => input_4bit(i),	
en data cnt => en data cnt	d(i),
en_addr_cnt => en_addr_cnt	(i),
MISO => MISO_d(i),	
NextState => NextState(i),	
NextState_vi => NextState_v NextState_rd -> NextState_r	d(i)):
en 4bit voter: voter	Voter unit
port map (voter_in1 => en_4bit_d(0),	
voter_in2 => en_4bit_d(1),	
voter_in3 => $en_4bit_d(2)$,	
voter_out => en_4bit_q(i));	 Votor unit
port map (voter in1 => en 8bit d(0)	
voter_in2 => en_8bit_d(1),	

V	voter_in3 => en_8bit_d(2),	
V	voter_out => en_8bit_q(i));	
en_data_cntr_voter: v port map (voter v v v	voter r_in1 => en_data_cnt_d(0), voter_in2 => en_data_cnt_d(1), voter_in3 => en_data_cnt_d(2), voter_out => en_data_cnt_q(i));	Voter unit
en_addr_cntr_voter: v	voter	Voter unit
port map (voter	r_in1 => en_addr_cnt_d(0),	
v	voter_in2 => en_addr_cnt_d(1),	
v	voter_in3 => en_addr_cnt_d(2),	
V FOL	voter_out => en_addr_cnt_q(i));	
end generate FSMs;		

end generate FSI end Stractural;

Κώδικας που περιλαμβάνει όλες τις μονάδες του slave (My_components.vhd):

use ieee.std_logic_	1164.all;			
package My_comp	onents is			
		SPI slave with S	Simple TMR	
Component s port (i_	spi_slave_TI _SCLK i_rstn i_CSN i_MOSI i_data_in o_wr_en o_MISO o_addr_ou	MR : in std_logic; : in std_logic; : in std_logic; : in std_logic; : in std_logic_vector : out std_logic; : out std_logic; t: out std_logic_vector t: out std_logic_vector	or (7 downto 0); ctor (3 downto 0);	Clock Reset Slave enable (low active) Serial input from master Data in from reg file for read mode Write enable Serial output to master Address for writing data to reg file Data out to reg file
end Compon	O_uala_ou			Data out to reg me
		SPI slave with		
Component s port (i_	spi_slave_fu SCLK1 i_CSN1 i_rstn1 i_MOSI1 i_data_in1 i_SCLK2 i_CSN2 i_rstn2 i_MOSI2 i_data_in2 i_SCLK3 i_CSN3 i_rstn3 i_CSN3 i_rstn3 i_data_in3 o_MISO1 o_wr_en1 o_addr_ou o_MISO2 o_wr_en2 o_addr_ou o_data_ou o_MISO3	<pre>II_TMR is : in std_logic; : out std_logic;</pre>	Clo Sla Re Se or (7 downto 0); I Clo Sla Re Se or (7 downto 0); I Clo Sla Re Se or (7 downto 0); I Se Wi ector (3 downto 0); ector (7 downto 0); ector (7 downto 0);	ock 1 ave enable (low active) 1 eset 1 Data in from reg file for read mode 1 ock 2 ave enable (low active) 2 eset 2 Data in from reg file for read mode 2 Data in from reg file for read mode 2 ock 3 ave enable (low active) 3 eset 3 erial input from master 3 Data in from reg file for read mode 3 erial input from master 1 rite enable 1 Addr for writing data to reg file 1 Serial output to master 2 Write enable 2 Addr for writing data to reg file 2 Addr for writing data to reg file 2 Data out to reg file 2 Serial output to master 3

end Compone	o_wr_en3 : out std_lo o_addr_out3: out std_lo o_data_out3: out std_lo ent;	gic; igic_vector (3 downto 0); gic_vector (7 downto 0));	Write enable 3Addr for writing data to reg file 3Data out to reg file 3
	Combina	tional part of FSM	
Component F port (M	SM_comb OSI : in std_log count_data : in std_log count_addr : in std_log data_in : in std_log	ic; ic_vector (3 downto 0); ic_vector (1 downto 0); ic_vector (7 downto 0);	 Serial input from master Counter for data Counter for address Data in from reg file (via register)
oddroco modo)	CurrState : in std_logi	c_vector (3 downto 0);	Output of state register (taking
	CurrState_wr: in std_log CurrState_rd: in std_log wr_en : out std_log en_4bit : out std_log input_4bit : out std_log input_8bit : out std_log en_data_cnt : out std_log MISO : out std_log NextState : out std_log NextState_wr: out std_log	gic_vector (1 downto 0); gic_vector (1 downto 0); gic; Wr c; Ena c; Dat ic; Dat ogic; Ena ogic; Ena ogic; Ena ogic; Sea gic_vector (3 downto 0); ogic_vector (1 downto 0); ogic_vector (1 downto 0);	Out of state reg (writing mode) Out of state reg (reading mode) en signal for reg file (via register) able for address shift register ta input for address shift register ta input for address shift register able for data shift register able for data counter able for address counter rial output, to master In of state reg (taking addr mode) In of state reg (reading mode)
	Shift rea	ister with TMR	
Component S generic	Shift_reg_TMR : (full_TMR: boolean := t bits : integer := clock : std_logic : Ports	rue; Set TMR 8; Data widt = '1'); Set clock.	implementaition. Simple or full h (Minimum 2) . 0: Falling, 1: Rising
end Compone	g_IIII : III std_logic, clk1 : in std_logic; rst1 : in std_logic; en1 : in std_logic; reg_in2 : in std_logic; clk2 : in std_logic; rst2 : in std_logic; reg_in3 : in std_logic; clk3 : in std_logic; rst3 : in std_logic; reg_out1: out std_logic; reg_out2: out std_logic_ reg_out3: out std_logic_ reg_out3: out std_logic_ reg_out3: out std_logic_ reg_out3: out std_logic_	_vector (bits-1 downto 0); _vector (bits-1 downto 0); _vector (bits-1 downto 0));	 Senai input 1 of shift register Clock 1, Rising edge Low active reset 1 High active enable 1 Serial input 2 of shift register Clock 2, Rising edge Low active reset 2 High active enable 2 Serial input 3 of shift register Clock 3, Rising edge Low active reset 3 High active enable 3 Parallel output of shift register Parallel output of shift register Parallel output of shift register
	Count	er with TMR	
Component C generic port (cl	Counter_TMR c (full_TMR: boolean := t bits : integer := clock : std_logic : Ports k1 : in std_log	rue; Set TMR 4; Bits of co = '1'); Set clock. ic;	implementaition. Simple or full unter (Minimum 2) . 0: Falling, 1: Rising Clock 1
	en1: in std_logrst1: in std_logclk2: in std_logen2: in std_logrst2: in std_logclk3: in std_log	ווכ; ווכ; ווכ; ווכ; ווכ;	Active high enable 1 Active low reset 1 Clock 2 Active high enable 2 Active low reset 2 Clock 3

en3 : in std_logic; rst3 : in std_logic; -- Active high enable 3 -- Active low reset 3 counter out1: out std logic vector (bits-1 downto 0); -- Counter output counter_out2: out std_logic_vector (bits-1 downto 0); -- Counter output counter out3: out std logic vector (bits-1 downto 0)); -- Counter output end Component; ----- TMR n-bit Register Component Generic TMR reg is

 generic (full_TMR: boolean := true;
 -- Set TMR implementaition. Simple or full

 bits : integer := 4;
 -- Data width (Minimum 2)

 clock : std_logic := '1');
 -- Set clock. 0: Falling, 1: Rising

 ------ Ports ----port (Data_in1 : in std_logic_vector (bits-1 downto 0); -- Data input 1 clk1 : in std_logic; -- Clock 1 rst1 : in std_logic; -- Reset 1 (active low) en1 : in std_logic; -- Enable 1 (active high) Data_in2 : in std_logic_vector (bits-1 downto 0); -- Data input 2 clk2 : in std_logic; -- Clock 2 rst2 : in std_logic; -- Reset 2 (active low) en2 : in std_logic; -- Enable 2 (active high) Data_in3 : in std_logic_vector (bits-1 downto 0); -- Data input 3 clk3 : in std logic; -- Clock 3 rst3 : in std logic; -- Reset 3 (active low) en3 : in std_logic;-- Enable 3 (activeData_out1: out std_logic_vector (bits-1 downto 0);-- TMR unit out 1Data_out2: out std_logic_vector (bits-1 downto 0);-- TMR unit out 2Data_out3: out std_logic_vector (bits-1 downto 0));-- TMR unit out 3 -- Enable 3 (active high) end Component; ----- TMR 1-bit Register Component TMR reg is generic (full_TMR: boolean := true; -- Set TMR implementaition. Simple or full clock : std_logic := '1'); -- Set clock. 0: Falling, 1: Rising Portsport (Data_in1 : in std_logic;-- Data input 1clk1 : in std_logic;-- Clock 1rst1 : in std_logic;-- Reset 1 (active low)en1 : in std_logic;-- Enable 1 (active high)Data_in2 : in std_logic;-- Data input 2clk2 : in std_logic;-- Clock 2rst2 : in std_logic;-- Clock 2rst2 : in std_logic;-- Enable 2 (active low)en2 : in std_logic;-- Enable 2 (active high)Data_in3 : in std_logic;-- Data input 3clk3 : in std_logic;-- Clock 3rst3 : in std_logic;-- Clock 3rst3 : in std_logic;-- Reset 3 (active low)en3 : in std_logic;-- Enable 3 (active high)Data_out1: out std_logic;-- TMR unit out 1Data_out2: out std_logic;-- TMR unit out 2Data_out3: out std_logic;-- TMR unit out 3Component;-- TMR unit out 3 end Component; ----- TMR's voter, 1-bit -----Component voter is oter_in1: in std_logic;-- Voter's input 1voter_in2: in std_logic;-- Voter's input 2voter_in3: in std_logic;-- Voter's input 3voter_out: out std_logic);-- Voter's out port (voter_in1: in std_logic; end Component; ----- TMR's voter, n-bit Component generic_voter is generic (bits: integer); -- Bits for each input (Minimum 2) ----- Ports -----Port (voter_in1: in std_logic_vector (bits-1 downto 0); -- Voter's input 1

voter_in2: in std_logic_vector (bits-1 downto 0);-- Voter's input 2voter_in3: in std_logic_vector (bits-1 downto 0);-- Voter's input 3voter_out: out std_logic_vector (bits-1 downto 0));-- Voter's out end Component; ----- D-FF ------Component dff is -- Set clock. 0: Falling, 1: Rising generic (clock: std_logic); ----- Ports -----Port (clk : in std_logic; -- Clock rst : in std_logic; -- Low active reset en : in std_logic; -- High active enable Din : in std_logic; -- Data input Dout : out std_logic); -- Data output end Component; ----- Generic Register Component Generic_Reg is eric (bits : integer := 8; -- Data width (Minimum 2) clock: std_logic := '1'); -- Set clock. 0: Falling, 1: Rising generic (bits : integer := 8; ----- Ports -----Port (clk : in std_logic; -- Clock rst : in std_logic; -- Low active reset en : in std logic; -- High active enable Din : in std logic vector (bits - 1 downto 0); -- Data input Dout : out std_logic_vector (bits - 1 downto 0)); -- Data output end Component;
----- Generic Shift Register ------Component Generic_Shift_reg is generic (bits : integer := 8; -- Data width (Minimum 2) clock: std_logic := '1'); -- Set clock. 0: Falling, 1: Rising ----- Ports -----Port (clk : in std logic; -- Clock rst : in std_logic; -- Low active reset en : in std_logic; -- High active enable reg_in : in std_logic; -- Serial input reg_out: out std_logic_vector (bits - 1 downto 0)); -- Parallel output end Component; ----- Register file ------Component register_file is port (i_clk : in std_logic; i_rstn : in std_logic; i_wr_en : in std_logic; -- Clock -- Reset

 i_istil_iogic,
 -- Reset

 i_wr_en
 : in std_logic;
 -- Write enabl

 i_data_in : in std_logic_vector (7 downto 0);
 -- Data input

 i_address : in std_logic_vector (3 downto 0);
 -- Address

 -- Write enable o_data_out: out std_logic_vector (7 downto 0)); -- Data output end Component;

end package My_components;

Ακολουθεί ο κώδικας των αρχείων για την προσομοίωση.

Κώδικας που υλοποιεί τον master σε επίπεδο προσομοίωσης (spi_stim_pkg.vhd):

library ieee; library std; use ieee.std_logic_1164.all; use ieee.std_logic_arith.all; use ieee.std_logic_signed.all; use ieee.std_logic_textio.all; use std.textio.all; package spi_stim_pkg is

-- Constants

```
constant hold_time : time := 1 ns;
                                                      -- Delay time
      constant SPI NAddrBits : integer := 4;
                                                      -- Number of data bits for address
      constant SPI_NDataBits : integer := 8:
                                                      -- Number of bits for data
      constant SPI_Polarity : boolean := false; -- false when CPOL = '0', true when CPOL = '1'
      constant SPI_Phase : boolean := false;
                                                      -- false when CPHA = '0', true when CPHA = '1'
       _____
      -- Access Procedures
       _____
      -- Write 1 or more data starting at specified address
      procedure spi write (
            signal addr : in
                             std logic vector (SPI NAddrBits-1 downto 0);
                                                                               -- Address input
            signal data : in std_logic_
signal clk : in std_logic;
                              std logic vector (SPI NDataBits-1 downto 0);
                                                                               -- Data input
                                                                         -- System clock
                                                                         -- Custom clock for slave
            signal sck : inout std_logic;
            signal csn : out std_logic;
                                                                         -- Slave enable
                                                                         -- Master serial out to slave
            signal mosi : out std_logic;
            signal miso : in
                             std_logic
                                                                         -- Master serial in from slave
     );
      -- Read 1 or more data starting at specified address
      procedure spi_read (
            signal addr : in
                              std logic vector (SPI NAddrBits-1 downto 0);
                                                                               -- Address input
            signal data : out std_logic_vector (SPI_NDataBits-1 downto 0);
                                                                               -- Data output
                             std logic;
                                                                         -- System clock
            signal clk : in
            signal sck : inoutstd logic:
                                                                         -- Custom clock for slave
            signal csn : out std logic;
                                                                         -- Slave enable
            signal mosi : out std_logic;
                                                                         -- Master serial out to slave
            signal miso : in std logic
                                                                         -- Master serial in from slave
     );
end package spi_stim_pkg;
package body spi stim pkg is
      procedure spi send address (
            signal addr : in std_logic_vector (SPI_NAddrBits-1 downto 0);
            signal clk : in
                              std logic;
            signal sck : inout std_logic;
            signal mosi : out std_logic;
            signal miso : in std_logic
      ) is
      begin
            for i in 0 to SPI NAddrBits-1 loop
                  -- wait for hold time;
                  mosi <= addr(SPI_NAddrBits-1-i);
                  if (SPI Phase) then
                        wait until falling edge(clk);
                        sck <= not sck:
                        wait until rising edge(clk);
                  else
                        wait until rising edge(clk);
                        sck <= not sck;
                        wait until falling_edge(clk);
                  end if:
                  sck <= not sck;
            end loop:
      end spi_send_address;
      procedure spi_send_data (
            signal data : in
                              std_logic_vector (SPI_NDataBits-1 downto 0);
            signal clk : in
                              std_logic;
            signal sck : inout std_logic;
            signal mosi : out std_logic;
                              std_logic
            signal miso : in
      ) is
```

```
begin
      for i in 0 to SPI NDataBits-1 loop
             -- wait for hold time;
             mosi <= data(SPI_NDataBits-1-i);
             if (SPI Phase) then
                    wait until falling_edge(clk);
                    sck <= not sck;
                    wait until rising edge(clk);
                    sck <= not sck;
             else
                    wait until rising edge(clk);
                    sck <= not sck;
                    wait until falling_edge(clk);
                    sck <= not sck;
             end if:
      end loop;
end spi_send_data;
procedure spi_receive_data (
      signal data : out std_logic_vector (SPI_NDataBits-1 downto 0);
      signal clk
                   : in
                          std_logic;
      signal sck : inout std_logic;
      signal mosi : out
                          std logic;
      signal miso : in
                          std_logic
) is
begin
      if ((data'length rem SPI NDataBits) /= 0) then
report "Wrong Data Size for SPI Reading (Should be multiple of SPI_NDataBits)." severity warning;
      end if;
      --wait for hold time;
      mosi <= '0';
      for i in 0 to (data'length)-1 loop
             if (SPI_Phase) then
                    wait until falling_edge(clk);
                    sck <= not sck;
                    wait for hold time;
                    if (miso = '0') then
                          data((data'length)-1-i) <= '0';</pre>
                    else
                           data((data'length)-1-i) \le '1';
                    end if;
                    wait until rising_edge(clk);
                    sck <= not sck;
             else
                    wait until rising_edge(clk);
                    sck <= not sck;
                    wait for hold time;
                    if (miso = '0') then
                           data((data'length)-1-i) \le '0';
                    else
                           data((data'length)-1-i) \le '1';
                    end if:
                    wait until falling edge(clk);
                    sck <= not sck;
             end if:
      end loop;
end spi_receive_data;
procedure spi_write (
      signal addr : in
                          std_logic_vector (SPI_NAddrBits-1 downto 0);
                                                                                -- Address input
      signal data : in
                          std_logic_vector (SPI_NDataBits-1 downto 0);
                                                                                -- Data input
      signal clk
                   : in
                          std_logic;
                                                                          -- System clock
                                                                          -- Custom clock for slave
      signal sck
                   : inoutstd_logic;
```

```
signal csn : out std_logic;
                                                                        -- Slave enable
      signal mosi : out std logic;
                                                                        -- Master serial out to slave
      signal miso : in
                         std logic
                                                                        -- Master serial in from slave
) is
begin
      if ((data'length rem SPI_NDataBits) /= 0) then
report "Wrong Data Size for SPI Writing (Should be multiple of SPI_NDataBits)." severity warning;
      end if:
      MOSI <= 'Z';
      sck <= 'Z';
      wait until falling edge(clk);
      csn <= '0';
      if (SPI Polarity) then
            sck <= '1';
      else
            sck <= '0';
      end if;
      if (SPI_Phase) then
            MOSI \le X';
            wait until rising_edge(clk);
            sck <= not sck;
            spi_send_address (addr,clk, sck, mosi, miso);
             -- wait for hold time;
            mosi <= '0';
            wait until falling edge(clk);
            sck <= not sck;
            wait until rising_edge(clk);
            sck <= not sck;
            spi send data (data, clk, sck, mosi, miso);
            wait until falling_edge(clk);
      else
            spi send address (addr,clk, sck, mosi, miso);
             -- wait for hold time;
            mosi <= '0';
            wait until rising_edge(clk);
            sck <= not sck;
            wait until falling_edge(clk);
            sck <= not sck;</pre>
            spi_send_data (data, clk, sck, mosi, miso);
            MOSI \le X';
            wait until rising_edge(clk);
      end if;
      csn <= '1';
      sck <= 'Z':
      MOSI <= 'Z':
      wait for 5 ns:
end spi_write;
procedure spi read (
      signal addr : in
                          std_logic_vector (SPI_NAddrBits-1 downto 0);
                                                                               -- Address input
      signal data : out std_logic_vector (SPI_NDataBits-1 downto 0);
                                                                               -- Data output
                                                                        -- System clock
      signal clk
                  : in
                         std logic;
      signal sck
                  : inoutstd logic;
                                                                        -- Custom clock for slave
      signal csn : out std_logic;
                                                                        -- Slave enable
      signal mosi : out std_logic;
                                                                        -- Master serial out to slave
      signal miso : in
                         std_logic
                                                                        -- Master serial in from slave
) is
begin
      if ((data'length rem SPI_NDataBits) /= 0) then
report "Wrong Data Size for SPI Reading (Should be multiple of SPI_NDataBits)." severity warning;
      end if;
```

```
MOSI \le 'Z';
             sck <= 'Z';
             wait until falling edge(clk);
             csn <= '0';
             if (SPI Polarity) then
                    sck <= '1';
             else
                    sck <= '0';
             end if:
             if (SPI_Phase) then
                    MOSI \le X';
                    wait until rising_edge(clk);
                    sck <= not sck;
                    spi send address (addr,clk, sck, mosi, miso);
                    -- wait for hold_time;
                    mosi <= '1';
                    wait until falling_edge(clk);
                    sck <= not sck;
                    wait until rising_edge(clk);
                    sck <= not sck;</pre>
                    spi_receive_data (data, clk, sck, mosi, miso);
                    wait until falling_edge(clk);
             else
                    spi send address (addr.clk, sck, mosi, miso);
                    -- wait for hold time:
                    mosi <= '1':
                    wait until rising_edge(clk);
                    sck <= not sck;
                    wait until falling_edge(clk);
                    sck <= not sck;
                    spi receive data (data, clk, sck, mosi, miso);
                    MOSI \le X';
                    wait until rising_edge(clk);
             end if:
             csn <= '1';
             sck <= 'Z';
             MOSI \le 'Z';
      end spi_read;
end package body spi_stim_pkg;
```

Κώδικας για την προσομοίωση της μονάδας των καταχωρητών με χρήση TMR (tb_generic_TMR_reg.vhd):

```
library ieee;
use ieee.std logic 1164.all;
use work.My_components.all;
entity tb generic TMR reg is

      generic (full_TMR : boolean := true;
      -- Set TMR implementaition. Simple or full

      bits : integer := 8;
      -- Data width (Minimum 2)

      clock : std_logic := '1');
      -- Set clock. 0: Falling, 1: Rising

end tb_generic_TMR_reg;
architecture tb of tb_generic_TMR_reg is
                               : std logic vector (bits-1 downto 0) := (others = '0');
        signal Data in1
        signal Data in2 : std logic vector (bits-1 downto 0) := (others => '0');
        signal Data in3 : std logic vector (bits-1 downto 0) := (others => '0');
        signal clk
                                : std_logic := '0';
                                : std_logic := '0';
        signal rst
                      : std_logic := '0';
        signal en
        signal Data_out1: std_logic_vector (bits-1 downto 0);
```

signal Data_out2: std_logic_vector (bits-1 downto 0); signal Data_out3: std_logic_vector (bits-1 downto 0); constant clk_period : time := 10 ns; -- Clock period definition begin -- Instantiate the Unit Under Test (UUT) uut: Generic_TMR_reg generic map (full_TMR => full_TMR, => bits, bits clock => clock) port map (Data_in1 => Data_in1, Data in2 => Data in2, Data_in3 => Data_in3, clk1 => clk,clk2 => clk,clk3 => clk,rst1 => rst, rst2 => rst, rst3 => rst, en1 => en, en2 => en, en3 => en, Data_out1 => Data_out1, Data_out2 => Data_out2, Data_out3 => Data_out3); -- Clock process definitions clk_process: process begin clk <= '0'; wait for clk_period/2; clk <= '1'; wait for clk period/2; end process; -- Stimulus process stim_proc: process begin wait for 15 ns; rst <= '0'; en <= '1'; wait for 15 ns; Data_in1 <= X"1C"; Data_in2 <= X"1C"; Data_in3 <= X"14"; wait for 15 ns; rst <= '1': Data in1 <= X"90"; Data_in2 <= X"90": Data_in3 <= X"99"; wait for clk_period; Data_in1 <= X"B1"; Data_in2 <= X"B1"; Data_in3 <= X"BB"; wait for clk_period; Data_in1 <= X"A3"; Data_in2 <= X"33"; Data_in3 <= X"A3"; wait for clk_period; Data_in1 <= X"DD"; Data_in2 <= X"DD"; Data_in3 <= X"33"; wait for clk_period; Data_in1 <= X"4A";

```
Data_in2 <= X"44";
      Data in 3 \le X'' 4A'';
      wait for clk_period;
      Data_in1 <= X"65":
      Data in2 <= X"78";
      Data in3 <= X"78";
      wait for clk_period;
      Data in1 <= X"55";
      Data in2 <= X"50";
      Data_in3 <= X"50";
      wait for clk period;
      Data in1 \leq X"F8";
      Data in2 <= X"53";
      Data_in3 <= X"B7";
      wait for 20 ns;
      report "End of simulation" severity failure;
      wait;
end process;
```

Κώδικας για την προσομοίωση του slave με υλοποίηση απλού TMR (tb_simple_ TMR spi slave.vhd):

library ieee; use ieee.std_logic_1164.all; use ieee.std_logic_arith.all; use ieee.std logic signed.all; use work.spi stim pkg.all; use work.My components.all; entity tb_simple_TMR_spi_slave is end entity tb_simple_TMR_spi_slave; architecture tb of tb_simple_TMR_spi_slave is constant clk_period : time := 10.0 ns; signal data_reg_to_spi : std_logic_vector (7 downto 0); signal data_spi_to_reg : std_logic_vector (7 downto 0); signal data_spi_to_reg : std_logic_vector (7 downto signal addr : std_logic_vector (3 downto 0); signal clk : std_logic := '0'; signal sclk : std_logic := '0'; signal CSN : std_logic := '1'; signal rstn : std_logic := '0'; signal MOSI : std_logic := '0'; signal wr_en: std_logic := '0'; signal MISO : std_logic; signal int_clk : std_logic; signal data_wr : std_logic_vector (7 downto 0); signal data_rd : std_logic_vector (7 downto 0); -- Master's data out signal addr1 : std_logic_vector (3 downto 0); begin int_clk <= sclk; dut_simple_TMR: spi_slave_TMR -- Simple TMR slave port map (i_SCLK => int_clk, -i CSN => CSN,-i_MOSI => MOSI, ----i rstn => rstn, i_data_in => data_reg_to_spi, -o_wr_en => wr_en, ---o addr out = addr, o data_out => data_spi_to_reg, -o MISO => MISO); ---

end tb;

```
dut_reg_file: register_file
                                                                    -- Register file
             port map (i_clk
                              => int clk,
                                                                    --
                          i rstn => rstn,
                                                                    --
                          i_wr_en => wr_en,
                                                                    --
                          i_data_in => data_spi_to_reg,
                                                                    ---
                          i_address => addr,
                                                                    --
                          o_data_out => data_reg_to_spi);
                                                                    --
 clk process :process
 begin
       clk <= '0';
       wait for clk_period/2;
       clk <= '1';
       wait for clk_period/2;
 end process;
 test_vectors: process is
 begin
       rstn <= '0';
       addr1 <= "0101";
       data_wr <= "00000001";
       wait for 40 ns;
       rstn <= '1';
       wait for 20 ns;
       ----- Write data ------
       addr1 <= "0001";
       data_wr <= "01010101":
       spi_write (addr1, data_wr, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
                   <= "0010";
       addr1
       data wr <= "00001111";
       spi_write (addr1, data_wr, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1 <= "0011";
       data_wr <= "11110000";
       spi_write (addr1, data_wr, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1 <= "0100";
       data_wr <= "10101010";
       spi_write (addr1, data_wr, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1
                   <= "1111";
       data_wr <= "11001001";
       spi_write (addr1, data_wr, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       ----- Read data ------
       addr1 <= "0001":
       spi_read (addr1, data_rd, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1 <= "0010";
       spi_read (addr1, data_rd, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1 <= "0011";
       spi_read (addr1, data_rd, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1 <= "0100";
       spi_read (addr1, data_rd, clk, sclk, CSN, MOSI, MISO);
       wait for 100 ns;
       addr1 <= "1111";
       spi_read (addr1, data_rd, clk, sclk, CSN, MOSI, MISO);
       wait for 120 ns;
report "End of simulation" severity failure;
```

end process test_vectors; end architecture tb;

Κώδικας για την προσομοίωση του slave με υλοποίηση πλήρους TMR (tb_full_ TMR_spi_slave.vhd):

```
librarv ieee:
use ieee.std logic 1164.all;
use ieee.std logic arith.all;
use ieee.std logic signed.all;
use work.spi_stim_pkg.all;
use work.My_components.all;
entity tb_full_TMR_spi_slave is
end entity tb full TMR spi slave;
architecture tb of tb full TMR spi slave is
       constant clk_period : time := 10.0 ns;
       type vector1 is array (2 downto 0) of std_logic_vector (3 downto 0);
       type vector2 is array (2 downto 0) of std_logic_vector (7 downto 0);
       signal data_reg_to_spi : vector2;
       signal data_spi_to_reg : vector2;
       signal addr
                         : vector1;
       signal clk
                          : std_logic := '0';
       signal sclk1 : std logic := '0';
       signal sclk2 : std logic := '0';
       signal sclk3 : std logic := '0';
       signal CSN1
                          : std logic := '1';
       signal CSN2
                          : std logic := '1';
       signal CSN3
                          : std logic := '1';
       signal rstn1 : std_logic := '0';
       signal rstn2 : std_logic := '0';
       signal rstn3 : std_logic := '0';
                         : std logic;
       signal MOSI1
                          : std_logic;
       signal MOSI2
      signal MOSI3
signal MISO1
signal MISO2
signal MISO3
signal data
                          : std_logic;
                          : std logic;
                          : std_logic;
                          : std_logic;
       signal data_wr1 : std_logic_vector (7 downto 0);
       signal data_rd1 : std_logic_vector (7 downto 0);
                                                                  -- Master's copy 1 data out
       signal data_rd2 : std_logic_vector (7 downto 0);
                                                                  -- Master's copy 2 data out
       signal data_rd3
                        : std_logic_vector (7 downto 0);
                                                                  -- Master's copy 3 data out
       signal addr1 : std_logic_vector (3 downto 0);
                          : std_logic_vector (2 downto 0) := "000";
       signal int_clk
       signal int rst: std logic vector (2 downto 0) := "000";
       signal wr en: std logic vector (2 downto 0) := "000";
       signal addr2 : std_logic_vector (3 downto 0) := "1111";
                                                                        -- Wrong data
       signal data wr2 : std logic vector (7 downto 0) := "11001100"; -- Wrong data
begin
       int clk <= sclk3 & sclk2 & sclk1;
      int rst <= rstn3 & rstn2 & rstn1;
       dut_Full_TMR_slave: spi_slave_full_TMR
                                                                        -- Full TMR slave
             port map (i_SCLK1 => int_clk(0),
                                                                        --
                                        => CSN1,
                                                                        ---
                           i CSN1
                                        => int_rst(0),
                            i_rstn1
                                                                        --
                           i_MOSI1 => MOSI1,
                                                                        ---
                           i_data_in1 => data_reg_to_spi(0),
                                                                        --
                            i\_SCLK2 => int\_clk(1),
                                                                        ---
                           i CSN2
                                        => CSN2,
```

```
i_rstn2
                               => int_rst(1),
                                                              ---
                   i MOSI2 => MOSI2,
                                                              ---
                   i_data_in2 => data_reg_to_spi(1),
                                                              --
                   i SCLK3 = int_clk(2),
                   i_CSN3 => CSN3.
                                                              ---
                              => int_rst(2),
                   i_rstn3
                                                              ---
                   i_MOSI3 => MOSI3,
                                                              ---
                   i_data_in3 => data_reg_to_spi(2),
                                                              --
                   o_MISO1 => MISO1,
o_wr_en1 => wr_en(0),
                                                              --
                                                              ---
                                                              --
                   o addr out1 => addr(0),
                   o_data_out1 => data_spi_to_reg(0),
                                                              --
                   o_MISO2 => MISO2,
o_wr_en2 => wr_en(1),
                                                              --
                                                              --
                   o_addr_out2 \Rightarrow addr(1),
                                                              ---
                   o_data_out2 => data_spi_to_reg(1),
                                                              --
                   o_MISO3 => MISO3,
                                                              ---
                   o_wr_en3 => wr_en(2),
                                                              --
                   o_addr_out3 => addr(2),
                                                              --
                    o_data_out3 => data_spi_to_reg(2));
                                                              --
dut_reg_files: for i in 2 downto 0 generate
      dut_reg_files: register_file
                                                              -- Register file
            port map (i_clk
                              => int_clk(i),
                                                              ---
                          i rstn => int_rst(i),
                                                              --
                          i wr en => wr en(i),
                                                              ---
                          i_data_in => data_spi_to_reg(i),
                                                              ---
                          i_address => addr(i),
                                                              --
                          o_data_out => data_reg_to_spi(i)); --
end generate dut reg files;
-- Clock process definitions
clk process :process
begin
      clk <= '0';
      wait for clk_period/2;
      clk <= '1';
      wait for clk_period/2;
end process clk_process;
Copy1: process is
begin
      rstn1 <= '0';
      addr1 <= "0101";
      data_wr1 <= "00000001";
      wait for 40 ns;
      rstn1 <= '1':
      wait for 20 ns;
      ----- Write data ------
      addr1 <= "1011":
      data wr1 <= "01010101";
      spi_write (addr1, data_wr1, clk, sclk1, CSN1, MOSI1, MISO1);
      wait for 100 ns;
                   <= "0010";
      addr1
      data_wr1 <= "00001111";
      spi_write (addr1, data_wr1, clk, sclk1, CSN1, MOSI1, MISO1);
      wait for 100 ns;
      addr1 <= "0011";
      data_wr1 <= "11110000";
      spi_write (addr1, data_wr1, clk, sclk1, CSN1, MOSI1, MISO1);
      wait for 100 ns;
      addr1 <= "0100";
      data_wr1 <= "10101010";
      spi_write (addr1, data_wr1, clk, sclk1, CSN1, MOSI1, MISO1);
```

```
wait for 100 ns;
               <= "1111";
       addr1
       data wr1 <= "11001001";
       spi_write (addr1, data_wr1, clk, sclk1, CSN1, MOSI1, MISO1);
       wait for 100 ns;
       ----- Read data ------
       addr1 <= "1011";
       spi_read (addr1, data_rd1, clk, sclk1, CSN1, MOSI1, MISO1);
       wait for 100 ns;
       addr1 <= "0010";
       spi read (addr1, data rd1, clk, sclk1, CSN1, MOSI1, MISO1);
       wait for 100 ns;
       addr1 <= "0011";
       spi_read (addr1, data_rd1, clk, sclk1, CSN1, MOSI1, MISO1);
       wait for 100 ns;
       addr1 <= "0100";
       spi_read (addr1, data_rd1, clk, sclk1, CSN1, MOSI1, MISO1);
       wait for 100 ns;
       addr1 <= "1111";
       spi_read (addr1, data_rd1, clk, sclk1, CSN1, MOSI1, MISO1);
       wait for 120 ns;
report "End of simulation" severity failure;
 end process copy1;
 Copy2: process is
 begin
       rstn2 <= '0';
       addr1 <= "0101":
       data wr1 <= "00000001";
       wait for 40 ns;
       rstn2 <= '1';
       wait for 20 ns;
       ----- Write data -----
       addr1 <= "1011";
       data wr1 <= "01010101";
       spi_write (addr1, data_wr1, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
               <= "0010";
       addr1
       data_wr1 <= "00001111";
       spi_write (addr2, data_wr2, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
       addr1 <= "0011";
       data_wr1 <= "11110000";
       spi write (addr1, data wr2, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
       addr1 <= "0100";
       data_wr1 <= "10101010":
       spi_write (addr2, data_wr1, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
       addr1 <= "1111";
       data_wr1 <= "11001001";
       spi_write (addr1, data_wr1, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
       ----- Read data ------
       addr1 <= "1011";
       spi_read (addr1, data_rd2, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
       addr1 <= "0010";
       spi_read (addr2, data_rd2, clk, sclk2, CSN2, MOSI2, MISO2);
       wait for 100 ns;
       addr1 <= "0011";
       spi_read (addr1, data_rd2, clk, sclk2, CSN2, MOSI2, MISO2);
```

```
wait for 100 ns;
            addr1 <= "0100";
            spi_read (addr2, data_rd2, clk, sclk2, CSN2, MOSI2, MISO2);
            wait for 100 ns;
            addr1 <= "1111";
            spi_read (addr1, data_rd2, clk, sclk2, CSN2, MOSI2, MISO2);
            wait for 120 ns;
    report "End of simulation" severity failure;
      end process copy2;
      Copy3: process is
      begin
            rstn3 <= '0';
            addr1 <= "0101";
            data wr1 <= "00000001";
            wait for 40 ns;
            rstn3 <= '1';
            wait for 20 ns;
            ----- Write data ------
            addr1
                       <= "1011";
            data_wr1 <= "01010101";
            spi_read (addr2, data_rd3, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
                         <= "0010":
            addr1
            data_wr1 <= "00001111";
            spi_write (addr1, data_wr1, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
            addr1 <= "0011";
            data wr1 <= "11110000";
            spi_write (addr2, data_wr1, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns:
            addr1 <= "0100";
            data_wr1 <= "10101010";
            spi_write (addr1, data_wr2, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns:
            addr1
                    <= "1111";
            data wr1 <= "11001001";
            spi_write (addr2, data_wr2, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
            ----- Read data ------
            addr1 <= "1011";
            spi_read (addr2, data_rd3, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
            addr1 <= "0010";
            spi_read (addr1, data_rd3, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
            addr1 <= "0011":
            spi_read (addr2, data_rd3, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
            addr1 <= "0100";
            spi_read (addr1, data_rd3, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 100 ns;
            addr1 <= "1111"
            spi_read (addr1, data_rd3, clk, sclk3, CSN3, MOSI3, MISO3);
            wait for 120 ns;
    report "End of simulation" severity failure;
      end process copy3;
end architecture tb;
```

ΑΝΑΦΟΡΕΣ

- [1] Jean-Claude Boudenot, Radiation Effects on Embedded Systems, *Springer-Dordrecht*, Radiation Space Environment, pp.1-9, 2007.
- [2] Ι. Βίβλης, Β. Ντούπης, «Επιδράσεις των ενεργητικών σωματιδίων της κοσμικής ακτινοβολίας στα μικροηλεκτρονικά συστήματα», Διπλωματική εργασία, τμήμα πληροφορικής και τηλεπικοινωνιών, ΕΚΠΑ, Κεφάλαιο 1.9, pp 18-19, 07/2008.
- [3] Remi Gaillard, Soft Errors in Modern Electronic Systems, Springer-Dordrecht, Chapter 2 Single Event Effects: Mechanisms and Classification, pp 27-54, 2011
- [4] F.W. Sexton, "Destructive single-event effects in semiconductor devices and ICs", IEEE Trans. Nucl. Sci. vol. 50, no. 3, Part 3, pp. 603–621, June 2003.
- [5] Scheick, Leif, Testing guideline for single event gate rupture (SEGR) of power MOSFETs, NASA Jet Propulsion Laboratory, p. 4, October 2008.
- [6] Γ. Πουϊκλής, «Ανάπτυξη Ψηφιακών Ηλεκτρονικών χαμηλής Ισχύος, σε ολοκληρωμένα κυκλώματα μικτού σήματος για διαστημικές εφαρμογές», Διδακτορική Διατριβή, τμήμα ηλεκτρολόγων μηχανικών και μηχανικών υπολογιστών, τομέας ηλεκτρονικής & συστημάτων πληροφορικής, Δημοκρίτειο Πανεπιστήμιο Θράκης, Κεφάλαιο 2, pp. 30-70, 2016.
- [7] F. W. Sexton, "Destructive Single Event Effects in semiconductor devices and silicon ICs", IEEE transactions on Nuclear Science, Vol. 50, No 3, pp. 603-621, 2003.
- [8] B. L. Gregory, και B. D. Shafer, "Latch up in CMOS integrated circuits", IEEE transactions on Nuclear Science, Vol. 20, pp. 2983-2991, 1973.
- [9] V. L. Pisacene και R. C. Moore, "Fundamentals of space systems", Oxford University Press, New York, 1994, chapter 12.
- [10] A. H. Johnston, Space radiation Effects on Microelectronics, JPL technical note.
- [11] E. P. Dodd, και L. W. Massengill, "Basic and modeling of single event upset in digital microelectronics", IEEE transactions on Nuclear Science, Vol. 50, No 3, pp. 583-602, 2003.
- [12] F. Faccio, K. Kloukinas, A. Marchioro, T. Calin, J. Cosculluela, M. Nicolaidis, και R. Velazco, "Single Event Effects in static and dynamic registers in a 0.25um CMOS technology", IEEE transactions on Nuclear Science, Vol. 46, pp. 1434-1439, 1999.
- [13] K. J. Hass, R. K. Treece και A. E. Giddings, "A radiation hardened 16/32-bit microprocessor", IEEE transactions on Nuclear Science, Vol. 36, pp. 2252-2257, Dec 1989
- [14] P.E. Dodd, "Device simulation of charge collection and single-event upset", IEEE Trans. Nucl. Sci., vol. 43, pp. 561–575, 1996.
- [15] S. Buchner et al, "Comparison of error rates in combinational and sequential logic", IEEE Trans. Nucl. Sci., vol. 44, no. 6, pp. 2209–2216, Dec. 1997.
- [16] R.C. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies", IEEE Trans. Device Mater. Reliab., vol. 5, no. 3, pp. 305–316, Sep. 2005.
- [17]T. R. Oldham, και F. B. McLean, "Total ionizing dose effects in MOS oxides and devices", IEEE transactions on Nuclear Science, Vol. 50, No. 3, pp. 483-499, June 2003
- [18] T. R. Oldham, "Ionizing radiation effects in MOS oxides, Advances in Solid state Electronics and Technology" (ASSET series), Singapore: World Scientific, 1999.
- [19]T. R. Oldham, και F. B. McLean, "Total ionizing dose effects in MOS oxides and devices", IEEE transactions on Nuclear Science, Vol. 50, No. 3, pp. 483-499, June 2003.
- [20] T. R. Oldham, A. J. Lelis, Jr H. E. Boesch, J. M. Benedetto, και F. B. McLean, "Post irradiation effects in field oxide isolation structures", IEEE transactions on Nuclear Science, Vol. 34, p. 1184, 1987.
- [21]M. R. Shaneyfelt, P. E. Dodd, B. L. Draper, και R. S. Flores, "Challenges in hardening technologies using shallow trench isolation", IEEE transactions on Nuclear Science, Vol. 45, p. 2584, 1998.
- [22] G. Aneli et. al. "Radiation tolerant VLSI circuits in standard deep submicron CMOS technologies for the LHC experiments: practical design aspects", IEEE Transactions on Nuclear Science, Vol. 46, pp.1690-1696, Dec 1999.
- [23] H. L. Hughes, και J. M. Benedetto, "Radiation effects and hardening of MOS technology: Devices and circuits", IEEE Transactions on Nuclear Science, vol. 50, pp.500-521, June 2003, 1999.
- [24] R. C. Lahoe, J. V. Osborn, R. Koga, S. Brown, και D.C. Mayer, "Application of hardness by design methodology to radiation tolerant ASIC technologies", IEEE Transactions on Nuclear Science, vol. 47, pp.2334, 2000.
- [25] M. Campbell et al., "A Pixel Readout Chip for 10-30 Mrad in Standard 0.25 µm CMOS", IEEE Transactions on Nuclear Science, Vol.46, No.3, p.156, June 1999
- [26] Federico Faccio, Radiation Effects on Embedded Systems, Springer-Dordrecht, Design Hardening Methodologies for ASICs, pp.143-160, 2007.
- [27] P.E. Dodd et al., "Impact of Technology Trends on SEU in CMOS SRAMs", IEEE Trans. Nucl. Science, Vol. 43, No. 6, p. 2797, December 1996.

- [28] Véronique Ferlet-Cavrois et al., Single Event Transients in Digital CMOS A Review, IEEE TRANSACTIONS ON NUCLEAR SCIENCE, vol. 60, No. 3, June 2013.
- [29] S. Niranjan, J.F. Frenzel, "A comparison of Fault-Tolerant State Machine Architectures for Space-Borne Electronics", IEEE Trans. On Reliability, Vol. 45, No. 1, p. 109, March 1996.
- [30] S. Lin, D.J. Costello Jr., "Error Control Coding", Second edition, Pearson Prentice Hall, 2004, ISBN 0-13-017973-6.
- [31] R. R. Troutman, Latchup in CMOS Technology: The Problem and its Cure. Boston, MA: Kulwer Publications, 1985.
- [32] N. A. Dodds, N. C. Hooten, R. A. Reed, "Effectiveness of SEL Hardening Strategies and the Latchup Domino Effect", IEEE Trans. Nucl. Sci., vol. 59, no. 6, pp. 2642-2650, Dec. 2012.
- [33] http://electronicsmentor.blogspot.com/process-corners.html. Προσπελάστηκε στις 18/12/2018.
- [34] http://siliconmentor.blogspot.com/process-corners.html. Προσπελάστηκε στις 18/12/2018.
- [35] Cadence, Virtuoso Liberate Reference Manual, Product version 15.1, August 2016.
- [36]<u>https://people.eecs.berkeley.edu/~alanmi/publications/other/liberty07_03.pdf</u>. Liberty Reference Manual (Version 2007.03), Προσπελάστηκε στις 20/12/2018.
- [37] https://sites.google.com/site/basic-char-terminology. Προσπελάστηκε στις 18/12/2018.
- [38] <u>https://vlsiuniverse.blogspot.com/2014/07/propagation-delay.html</u>. Προσπελάστηκε στις 18/12/2018.
- [39] Rohit Sharma, Characterization and modeling of digital circuits, 2016.
- [40] <u>https://www.digikey.com/eewiki/pages/pageld=7569477</u>. Προσπελάστηκε στις 12/1/2019.
- [41]<u>https://www.corelis.com/education/tutorials/spi-tutorial/</u>. Προσπελάστηκε στις 12/1/2019.
- [42] Sandi Habinc, "Functional Triple Modular Redundancy (FTMR)", Gaisler research, pp 6-8, Dec 2002.