

Ανάπτυξη κάρτας PCI διαύλου I²C για τον έλεγχο ηλεκτρονικών κυκλωμάτων FE σε ανιχνευτές πυριτίου.

169

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΓΕΩΡΓΙΟΣ Χ. ΣΙΔΗΡΟΠΟΥΛΟΣ

ΗΛΕΚΤΡΟΝΙΚΟΣ ΜΗΧΑΝΙΚΟΣ ΤΕ

Δ ΗΠΛΕ

Μ.Ε. 2

ΕΠΙΒΛΕΨΗ: ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ Ν. ΜΑΝΘΟΣ
ΕΡΓΑΣΤΗΡΙΟ ΦΥΣΙΚΗΣ ΗΨΗΛΩΝ ΕΝΕΡΓΕΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΣΤΙΣ ΣΥΓΧΡΟΝΕΣ ΗΛΕΚΤΡΟΝΙΚΕΣ ΤΕΧΝΟΛΟΓΙΕΣ

ΤΜΗΜΑ ΦΥΣΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΙΩΑΝΝΙΝΑ ΣΕΠΤΕΜΒΡΙΟΣ 2002

Χρ. 546

3

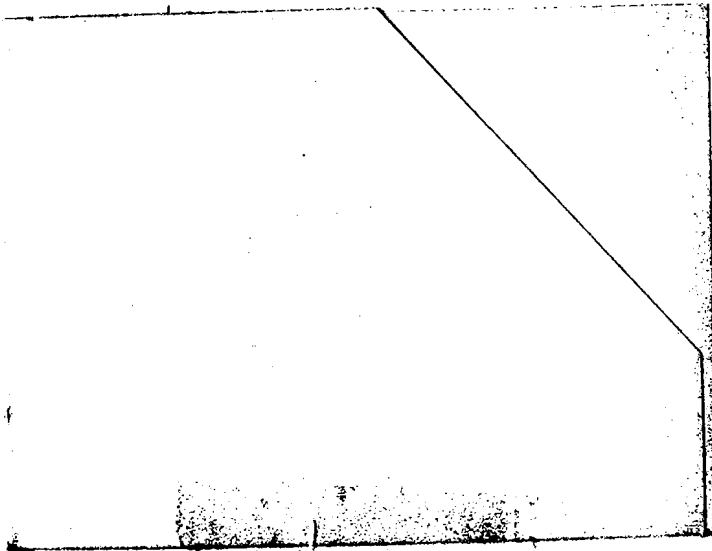


ΒΙΒΛΙΟΘΗΚΗ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΙΩΑΝΝΙΝΩΝ



026000132982

47/2009

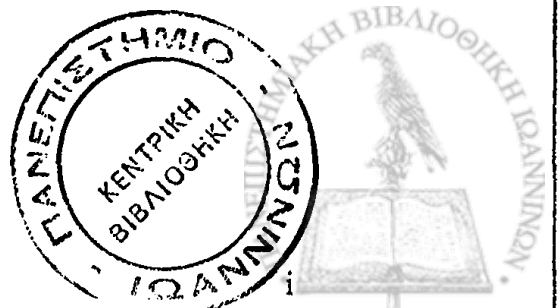


Περίληψη

Η παρούσα διπλωματική εργασία, έχει ως αντικείμενο την ανάπτυξη μίας κάρτας PCI, για τον έλεγχο των αναλογικών ηλεκτρονικών Front-End που χρησιμοποιούνται σε μικρολωριδιακούς ανιχνευτές πυριτίου. Τα σήματα ελέγχου των αναλογικών ηλεκτρονικών FE για ανιχνευτές πυριτίου, ακολουθούν συνήθως το καθιερωμένο πρωτόκολλο σειριακής επικοινωνίας I²C.

Η κάρτα PCI βασίζεται σε 2 ολοκληρωμένα κυκλώματα, το PCF8584 της εταιρίας PHILIPS και το S5920 της AMMC. Το ολοκληρωμένο PCF8584 είναι ένας ελεγκτής για το δίαυλο I²C ενώ το S5920 είναι μια γέφυρα διασύνδεσης με το δίαυλο PCI.

Η κάρτα αποτελεί μέρος ενός γενικότερου ηλεκτρονικού συστήματος που βασίζεται σε ένα προσωπικό Η/Υ και έχει ως στόχο την καταγραφή και επεξεργασία της πληροφορίας που συλλέγεται από αισθητήρες πυριτίου.



Abstract

The subject of the present MSc thesis is the development of a PCI card which controls the analog front-end electronics used on microstrip silicon detectors. The signals used to control the analog front-end chips usually follow the I²C serial communication protocol.

The PCI card is based on two integrated circuits, the PCF8584 manufactured by Philips and the S5920 manufactured by AMCC. PCF8584 is an I²C bus Controller and S5920 is a PCI target Interface.

This add-in card is going to be integrated into a personal computer based DAQ system, which will be used to acquire and process the information derived from microstrip silicon detectors.



Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέβαλαν σημαντικά στην πραγματοποίηση αυτής της εργασίας.

Τον επιβλέποντα της διπλωματικής μου εργασίας Επίκουρο Καθηγητή Ν. Μάνθο, μέλος του Εργαστηρίου Φυσικής Υψηλών Ενεργειών του Πανεπιστημίου Ιωαννίνων, για την καθοδήγησή του καθ' όλη τη διάρκεια εκπόνησης της εργασίας μου.

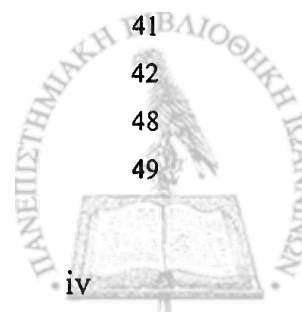
Τον Καθηγητή Φ. Τριάντη διευθυντή του ΕΦΥΕ, καθώς και τα άλλα μέλη του ΕΦΥΕ Επίκουρους Καθηγητές Π. Κόκκα και Ι. Ευαγγέλου για τη βοήθεια που μου προσέφεραν και την εμπιστοσύνη που μου έδειξαν.

Τους φίλους και συναδέλφους κ Π. Βηχούδη, για την άριστη συνεργασία σε όλο αυτό το διάστημα, και κ Α. Ασημίδη, για τη μύηση στα μυστικά του PCI.



Περιεχόμενα

	Εισαγωγή	1
1.	Εισαγωγή στο I²C	3
	1.1 Ιστορικά	3
	1.2 Η ιδέα του I2C	3
	1.3 Γενικά χαρακτηριστικά	5
	1.4 Διαδικασίες START και STOP	6
	1.5 Μεταφορά δεδομένων	7
	1.5.1 Δομή Δεδομένων (ByteFormat)	7
	1.5.2 Επιβεβαίωση	7
	1.6 Arbitration	8
	1.7 Διευθυνσιοδότηση	9
	1.7.1 Διευθυνσιοδότηση 7 bits	10
	1.7.2 Διευθυνσιοδότηση 10 bits	11
2.	Εισαγωγή στο PCI	15
	2.1 Ο διάυλος PCI και η ιστορία των διαύλων στους προσωπικούς Η/Υ	15
	2.2 Γενικά χαρακτηριστικά του PCI	17
	2.3 Βασικές έννοιες στο διάυλο PCI	20
	2.4 Συσκευές PCI μιας ή πολλών λειτουργιών.	21
	2.5 Οι ομάδες σημάτων του διαύλου PCI	21
	2.6 Τύποι προσαρτώμενων καρτών PCI	23
	2.7 Περιοχή ρυθμίσεων	24
	2.8 Μεταγωγή Ανακλώμενου Σήματος	27
	2.9 Χρονικά διαγράμματα του διαύλου PCI	28
3.	AMCC S5920 PCI Bridge	31
	3.1 Γενική περιγραφή του S5920	31
	3.2 Λειτουργία PASS-THRU	32
	3.3 Λειτουργία MAILBOX	34
	3.3.1 Περιγραφή Λειτουργίας	35
	3.3.2 Ένδειξη πληρότητας του Γραμματοκιβωτίου	36
	3.3.3 Εναλλακτική πρόσβαση καταχωρητών Mailbox	37
	3.4 Καταχωρητές λειτουργίας του S5920	38
4.	Γέφυρες I2C	41
	4.1. Γενική περιγραφή του PCF8584	41
	4.2 Καταχωρητές	42
	4.3 Χρονικά διαγράμματα	48
	4.4 Software	49

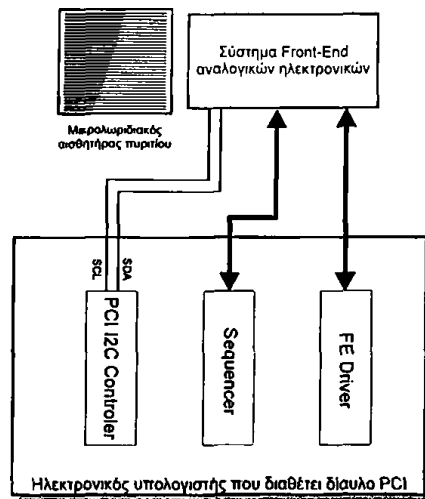


5.	RS-232 I2C Controller	
5.1	Σκοπός Υλοποίησης	53
5.2	Hardware	53
5.2.1	Βαθμίδα Ελέγχου	54
5.2.2	Βαθμίδα διεπαφής με τη θύρα RS-232	55
5.2.3	Βαθμίδα παραγωγής παλμού ρολογιού	55
5.3	Software	56
5.3.1	Έλεγχος του I2C Controller μέσω Hyper Terminal	56
5.3.2	Έλεγχος του I2C Controller μέσω LabVIEW	57
5.3.2.1	Εισαγωγή	57
5.3.2.2	Το virtual instrument της συσκευής i2c.vi	58
6.	Κάρτα PCI	63
6.1	Γενικά	63
6.2	Βαθμίδα διασύνδεσης με το δίαυλο PCI	64
6.3	Βαθμίδα σημάτων I2C	66
6.4	Βαθμίδα οπτικής απομόνωσης και προσαρμογής επιπέδων	68
6.4.1	Οπτική απομόνωση και προσαρμογή επιπέδων με χρήση οπτοζευκτών.	68
6.4.2	Προσαρμογή επιπέδων με χρήση του OK NE3372	70
6.5	Διάταξη των ακίδων του συνδέσμου της κάρτας.	71
6.6	Τοποθέτηση των εξαρτημάτων στην κάρτα PCI	71
6.7	Οδήγηση της κάρτας PCI	74
6.8	Δοκιμή της κάρτας PCI σε περιβάλλον DOS	76
6.9	Αποτίμηση της κάρτας PCI	78
	Αναφορές – Βιβλιογραφία	81
	ΠΑΡΑΡΤΗΜΑ Α	83
	Φωτογραφίες της κάρτας PCI	83
	ΠΑΡΑΡΤΗΜΑ Β	85
	Ηλεκτρονικά σχέδια και τυπωμένα κυκλώματα	85
	Ηλεκτρονικό σχέδιο κάρτας PCI.	86
	Τυπωμένο κύκλωμα κάρτας PCI.	88
	Τοποθέτηση εξαρτημάτων στην κάρτα PCI.	88
	Κατάλογος υλικών της κάρτας PCI.	90
	Ηλεκτρονικό σχέδιο RS-232 I ² C Controller	91
	Τυπωμένο κύκλωμα RS-232 I ² C Controller	92
	Τοποθέτηση εξαρτημάτων στο τυπωμένο του RS-232 I ² C Controller	92
	Κατάλογος υλικών του RS-232 I ² C Controller	92
	ΠΑΡΑΡΤΗΜΑ Γ	93
	Προγραμματισμός του μικροελεγκτή σε γλώσσα Assembly	93
	ΠΑΡΑΡΤΗΜΑ Δ	107
	Τα σήματα του διαύλου PCI.	107



Εισαγωγή

Στο εργαστήριο φυσικής υψηλών ενεργειών του πανεπιστημίου Ιωαννίνων, αναπτύσσεται ένα σύστημα ελέγχου μικρολωριδιακών αισθητήρων πυριτίου – αναλογικών ηλεκτρονικών. Σκοπός του συστήματος αυτού, είναι ο έλεγχος, η καταγραφή και η επεξεργασία της πληροφορίας που παράγεται στους αισθητήρες κατά τη διέλευση φωτονίων ή φορτισμένων σωματιδίων. Το σύστημα βασίζεται στο δίαυλο PCI και φιλοξενείται στο εσωτερικό ενός προσωπικού ηλεκτρονικού υπολογιστή.



Το σύστημα περιλαμβάνει τις εξής κάρτες PCI :

- Sequencer

Η κάρτα αυτή παρέχει τα αναγκαία σήματα χρονισμού για το σύστημα αναλογικών ηλεκτρονικών FE (Front End).

- Front End Driver

Η κάρτα αυτή, λαμβάνει την αναλογική έξοδο του FE και τη μετατρέπει σε ψηφιακή πληροφορία, την οποία στη συνέχεια επεξεργάζεται ο Η/Υ στον οποίο φιλοξενείται η κάρτα.

- I²C Controller

Η κάρτα αυτή, παρέχει τα απαραίτητα σήματα ελέγχου και ρύθμισης του συστήματος FE και αποτελεί το αντικείμενο ανάπτυξης της παρούσης διπλωματικής εργασίας.

Τα σήματα ελέγχου που παράγει η κάρτα PCI, υπόκεινται στο πρωτόκολλο σειριακής επικοινωνίας I²C. Πρόκειται για ένα πρωτόκολλο σύγχρονης σειριακής επικοινωνίας, το οποίο έχει υιοθετηθεί από τους περισσότερους κατασκευαστές ηλεκτρονικών εξαρτημάτων, ως ο απλούστερος δίαυλος διασύνδεσης των

ολοκληρωμένων κυκλωμάτων μεταξύ τους. Λόγω της διάδοσης του πρωτοκόλλου I²C, η κάρτα PCI μπορεί να χρησιμοποιηθεί και σε άλλες εφαρμογές.

Η κάρτα PCI βασίζεται σε 2 ολοκληρωμένα κυκλώματα, το PCF8584 της εταιρίας PHILIPS και το S5920 της AMCC. Το ολοκληρωμένο PCF8584, είναι ένας ελεγκτής για το δίαυλο I²C, ενώ το S5920 είναι μια γέφυρα διασύνδεσης με το δίαυλο PCI.

Στα πλαίσια της διπλωματικής εργασίας, υλοποιήθηκε και μία αυτόνομη μονάδα παραγωγής σημάτων I²C, η οποία συνδέεται με ένα Η/Υ μέσω της σειριακής θύρας RS-232. Ο λόγος που υλοποιήθηκε η μονάδα αυτή, ήταν η δοκιμή του ολοκληρωμένου PCF8584, πριν την τελική χρησιμοποίηση του στην κάρτα PCI.

Στα 2 πρώτα κεφάλαια που ακολουθούν, γίνεται μια σύντομη αλλά περιεκτική αναφορά στους διαύλους I²C και PCI. Ακολουθεί η ανάλυση της λειτουργίας των ολοκληρωμένων S5920 και PCF8584 στα κεφάλαια 3 και 4 αντίστοιχα. Στα κεφάλαια 5 και 6, γίνεται ανάλυση της αυτόνομης μονάδας I²C και της κάρτας PCI.

Τέλος, αξίζει να σημειωθεί ότι τόσο η σχεδίαση, όσο και η κατασκευή του τυπωμένου κυκλώματος της κάρτας PCI, πραγματοποιήθηκε εξ' ολοκλήρου στο Εργαστήριο Φυσικής Υψηλών Ενεργειών του Πανεπιστημίου Ιωαννίνων, χωρίς τη συμβολή κάποιας βιομηχανίας κατασκευής τυπωμένων κυκλωμάτων.

Συμβάσεις κειμένου

Η παράγραφος αυτή περιγράφει τους τυπογραφικούς συμβολισμούς που χρησιμοποιούνται σε αυτό το βιβλίο

Δεκαεξαδικοί αριθμοί

Όλοι οι δεκαεξαδικοί αριθμοί ακολουθούνται από το γράμμα **h** π.χ. A90Ch

Δυαδικοί αριθμοί

Όλοι οι δυαδικοί αριθμοί ακολουθούνται από το γράμμα **b** π.χ. 0100b

Δεκαδικοί αριθμοί

Όλοι οι δεκαδικοί αριθμοί ακολουθούνται από το γράμμα **d** ή μένουν ως έχουν π.χ. 127d ή 127

Ονομασία σημάτων

Κάθε ψηφιακό σήμα που ενεργοποιείται με χαμηλή λογική κατάσταση συμβολίζεται με το σύμβολο (#) π.χ. TRDY#



1. Εισαγωγή στο I²C

1.1 Ιστορικά

Το πρωτόκολλο I²C (Inter Integrated Circuit) προτάθηκε και αναπτύχθηκε το 1982 από την εταιρία Philips ως ένα πρωτόκολλο το οποίο επιτρέπει σε ολοκληρωμένα κυκλώματα να συνδέονται μεταξύ τους, μέσω ενός διαύλου σειριακής επικοινωνίας [1].

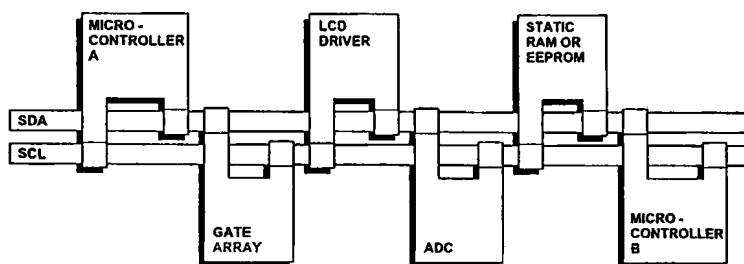
- Οι πρώτες εφαρμογές του I²C ήταν σε συσκευές τηλεόρασης όπου μια CPU συνδεόταν μέσω I²C με διάφορα ολοκληρωμένα κυκλώματα όπως ψηφιακά ποτενσιόμετρα, κυκλώματα επεξεργασίας χρώματος, φωτεινότητας και άλλα. Σήμερα μια πληθώρα εταιριών έχουν υιοθετήσει το πρωτόκολλο I²C και κατασκευάζουν μια μεγάλη ποικιλία από ημιαγωγικά στοιχεία όπως μνήμες ROM και EPROM, μετατροπείς ADC και DAC, ειδικούς ανιχνευτές και αισθητήρες και πολλά άλλα που χρησιμοποιούν το εν λόγω πρωτόκολλο.

Το I²C υποστήριξε αρχικά τη διασύνδεση μεταξύ ολοκληρωμένων κυκλωμάτων με ταχύτητες μεταφοράς δεδομένων των 100 Kbit/s. Στη συνέχεια η ταχύτητα μεταφοράς δεδομένων αυξήθηκε στα 400 Kbit/s (fast mode I²C) και σήμερα φτάνει τα 3,4 Mbit/s (High speed Mode I²C).

1.2 Η ιδέα του I²C

Ο δίαυλος I²C υποστηρίζει όλες τις τεχνολογίες κατασκευής ολοκληρωμένων κυκλωμάτων (NMOS, CMOS, Bipolar). Δύο αγωγοί, οι serial data (SDA) και serial clock (SCL), χρησιμοποιούνται για τη μεταφορά των πληροφοριών μεταξύ των στοιχείων που είναι συνδεδεμένα στο δίαυλο. Κάθε μονάδα-συσκευή, που είναι συνδεδεμένη στο δίαυλο, αναγνωρίζεται από μια μοναδική διεύθυνση (είτε είναι μικροελεγκτής ή οδηγός οθόνης LCD ή μνήμη ή διεπαφή ηλεκτρολογίου) και μπορεί να λειτουργήσει είτε ως πομπός, είτε ως δέκτης, ανάλογα με τη λειτουργία του. Προφανώς ένας οδηγός οθόνης LCD αποτελεί μόνο δέκτη, ενώ μια μνήμη μπορεί να εκπέμπει ή να λαμβάνει δεδομένα. Εκτός από το διαχωρισμό σε πομπούς και δέκτες οι μονάδες που συνδέονται στο δίαυλο I²C μπορούν να διαχωριστούν σε «αφέντες» (masters) και «σκλάβους» (slaves).

Μια συσκευή master είναι η μονάδα εκείνη που ξεκινά μια μεταφορά δεδομένων στο δίαυλο και δημιουργεί τα απαραίτητα σήματα συγχρονισμού (clock) για να επιτραπεί η μετάδοση. Σε αυτή τη φάση οποιαδήποτε συσκευή διευθυνσιοδοτείται, θεωρείται συσκευή slave. Ο δίαυλος I²C υποστηρίζει πολλές συσκευές master. Αυτό σημαίνει ότι μπορούν να συνδεθούν περισσότερες από μια συσκευές που μπορούν να ελέγξουν το δίαυλο. Συσκευές master αποτελούν συνήθως οι μικροελεγκτές (μC) ή μικροεπεξεργαστές (με έναν όρο μικροϋπολογιστές).



Σχήμα 1-1. Παράδειγμα ενός διαύλου I²C στον οποίο υπάρχουν 2 μικροελεγκτές

Το σχήμα 1-1 αντικατοπτρίζει τις σχέσεις μεταξύ master-slave και πομπό-δέκτη που μπορούν να απαντηθούν στο δίαυλο I²C. Πρέπει να σημειωθεί ότι οι τελευταίες σχέσεις, δεν είναι μόνιμες και εξαρτώνται μόνο από την κατεύθυνση των δεδομένων τη συγκεκριμένη στιγμή.

Η μεταφορά δεδομένων μπορεί να έχει ως εξής:

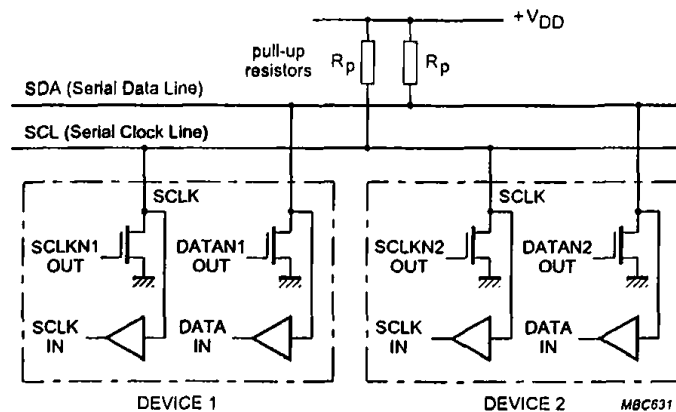
- 1) Υποθέτουμε ότι ο μC A επιθυμεί να στείλει πληροφορίες στον μC B:
 - Ο μC A (master), απευθύνεται (διευθυνσιοδοτεί) στον μC B (slave).
 - Ο μC A (master-πομπός) αποστέλλει δεδομένα στον μC B (slave-αποδέκτης)
 - Ο μC A τερματίζει τη διαδικασία μεταφοράς
- 2) Αν ο μC A επιθυμεί να λάβει πληροφορίες από τον μC B:
 - Ο μC A (master) απευθύνεται στον μC B (slave)
 - Ο μC A (master-αποδέκτης) λαμβάνει δεδομένα από τον μC B (slave-πομπός)
 - Ο μC A τερματίζει τη διαδικασία μεταφοράς

Ακόμη και στην περίπτωση αυτή, ο master (μC A) δημιουργεί τα απαραίτητα σήματα συγχρονισμού και τερματίζει την όλη διαδικασία.

Η πιθανότητα ύπαρξης περισσοτέρων του ενός μικροελεγκτών στο δίαυλο σημαίνει ότι περισσότεροι του ενός master θα προσπαθήσουν να ξεκινήσουν μια διαδικασία μεταφοράς την ίδια χρονική στιγμή. Για να αποφευχθεί μια χαώδης κατάσταση που θα προκύψει από μια τέτοια πιθανότητα, έχει αναπτυχθεί μια μέθοδος διαίτησίας (arbitration). Η μέθοδος αυτή βασίζεται στη διασύνδεση τύπου wired AND με την οποία συνδέονται όλες οι διεπαφές των στοιχείων του διαύλου I²C.

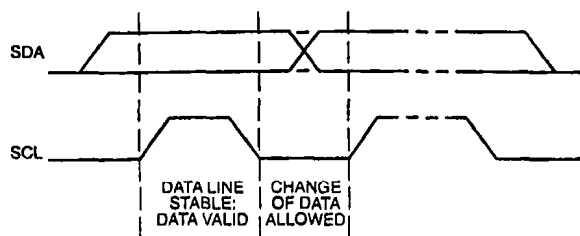
1.3 Γενικά χαρακτηριστικά

Οι επαφές SDA και SCL είναι αμφίδρομες γραμμές διασύνδεσης και συνδέονται σε μια θετική τάση μέσω μια πηγής ρεύματος ή μέσω ενός αντιστάτη "pull-up" (βλέπε σχήμα 1-2). Όταν ο δίαυλος είναι ελεύθερος και οι δύο γραμμές βρίσκονται σε λογικό "1". Τα στάδια εξόδου των διατάξεων που συνδέονται στον δίαυλο πρέπει να είναι τύπου ανοικτού συλλέκτη ή ανοικτού απαγωγού για να σχηματίζεται η συνολική διασύνδεση τύπου wired - AND. Το πλήθος των μονάδων που μπορούν να συνδεθούν στο δίαυλο εξαρτάται μόνο από το όριο χωρητικότητας του διαύλου το οποίο είναι 400 pF.



Σχήμα 1-2. Σύνδεση συσκευών στο δίαυλο I²C

Η τεχνική ανοικτού συλλέκτη ή ανοικτού απαγωγού έχει ένα μειονέκτημα. Στην περίπτωση που δίαυλος έχει μεγάλο μήκος, περιορίζεται η ταχύτητα μετάδοσης που μπορεί να επιτευχθεί. Οι μεγάλες γραμμές μεταφοράς προσδίδουν ένα χωρητικό φορτίο στην έξοδο. Στην περίπτωση που χρησιμοποιείται μια αντίσταση pull up δημιουργείται μια σταθερά χρόνου RC που θα έχει ως αποτέλεσμα τη δημιουργία ανακλάσεων στο σήμα. Όσο αυξάνεται η τιμή της σταθεράς χρόνου RC τόσο μειώνεται η τιμή της συχνότητας στην οποία μπορεί να λειτουργήσει σωστά ο δίαυλος.



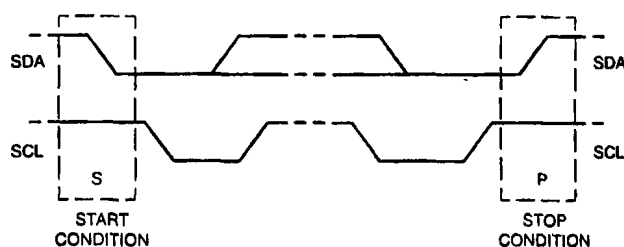
Σχήμα 1-3. Επιτρεπτές μεταβάσεις στο δίαυλο I²C.

Τα δεδομένα στη γραμμή SDA πρέπει να διατηρούνται σταθερά κατά την υψηλή κατάσταση της περιόδου του ρολογιού. Η υψηλή ('1') ή χαμηλή ('0') κατάσταση της γραμμής δεδομένων SDA μπορεί να αλλάξει όταν το σήμα ρολογιού στη γραμμή SCL είναι σε χαμηλή λογική κατάσταση (βλ. σχήμα 1-3).

1.4 Διαδικασίες START και STOP

Πριν από κάθε διαδικασία στον δίαυλο I²C πρέπει να προηγηθεί μια διαδικασία που ονομάζεται START. Μια μετάβαση από υψηλή λογική κατάσταση σε χαμηλή στη γραμμή SDA, καθώς η γραμμή SCL βρίσκεται σε υψηλή λογική κατάσταση, ονομάζεται «Διαδικασία έναυσης» (START CONDITION).

Ομοίως μια μετάβαση, από χαμηλή λογική κατάσταση σε υψηλή στη γραμμή SDA, καθώς η γραμμή SCL βρίσκεται σε υψηλή λογική κατάσταση, ονομάζεται «Διαδικασία τερματισμού» (STOP CONDITION).



Σχήμα 1-4. Διαδικασίες START και STOP.

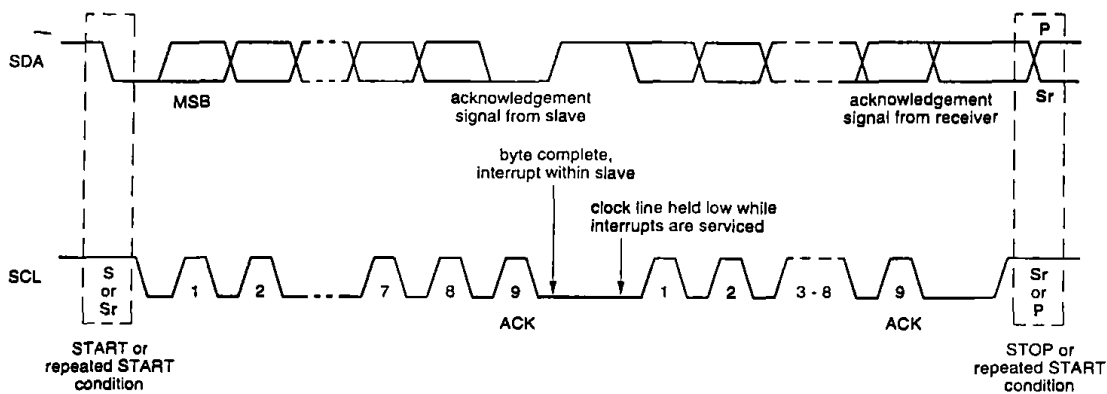
Οι δύο αυτές διαδικασίες εικονίζονται στο παραπάνω σχήμα 1-4. Οι διαδικασίες START (S) και STOP (P) παράγονται πάντα από τους master. Μετά το πέρας μιας κατάστασης START, ο δίαυλος θεωρείται απασχολημένος. Ο δίαυλος απελευθερώνεται πάλι μετά από μια διαδικασία STOP.

Ο δίαυλος παραμένει απασχολημένος μετά από μια επαναληπτική διαδικασία START (Sr) αντί μιας διαδικασίας STOP.

1.5 Μεταφορά δεδομένων

1.5.1 Δομή Δεδομένων (Byte Format)

Κάθε πληροφορία που αποστέλλεται (σειριακά) στη γραμμή SDA πρέπει να έχει μήκος 8 bits (ενός byte). Ο αριθμός των byte που μπορούν να αποσταλούν στα πλαίσια μιας μετάδοσης είναι απεριόριστος. Κάθε byte πρέπει να ακολουθείται από ένα bit επιβεβαίωσης (Acknowledge bit) (βλέπε § 1.5.2). Τα δεδομένα αποστέλλονται ξεκινώντας από το περισσότερο σημαντικό bit (MSB) (βλέπε σχήμα 1-5).



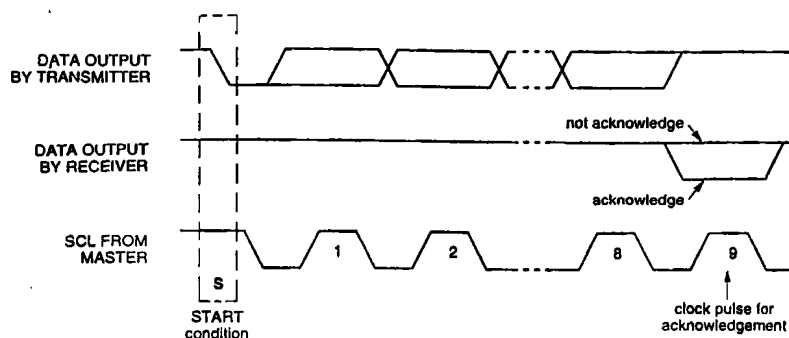
Σχήμα 1-5. Μεταφορά δεδομένων στο δίαυλο I²C.

Αν μια συσκευή Slave δεν μπορεί να λάβει ή να μεταδώσει δεδομένα επειδή είναι απασχολημένη με κάποια άλλη διαδικασία (π.χ. η εξυπηρέτηση ενός interrupt) μπορεί να κρατήσει τη γραμμή SCL σε χαμηλή λογική, κατάσταση αναμονής (wait state). Η μεταφορά δεδομένων συνεχίζεται όταν ο slave είναι έτοιμος για ένα ακόμη byte δεδομένων, οπότε και απελευθερώνει τη γραμμή SCL.

1.5.2 Επιβεβαίωση

Η μεταφορά δεδομένων με επιβεβαίωση είναι υποχρεωτική. Ο παλμός ρολογιού που συσχετίζεται με το bit επιβεβαίωσης παράγεται από τον master. Ο πομπός απελευθερώνει τη γραμμή SDA κατά τη διάρκεια του παλμού ρολογιού για το bit επιβεβαίωσης.

Ο δέκτης πρέπει να κρατήσει τη γραμμή SDA σε χαμηλή κατάσταση κατά τη διάρκεια του αντίστοιχου παλμού ρολογιού (βλέπε σχήμα 1-6). Βέβαια, οι χρόνοι ετοιμασίας και συγκράτησης (set up and hold) πρέπει να ληφθούν υπόψη. Συνήθως ο δέκτης που έχει διευθυνσιοδοτηθεί πρέπει να παράγει έναν παλμό ACK μετά την περάτωση μεταφοράς ενός byte.



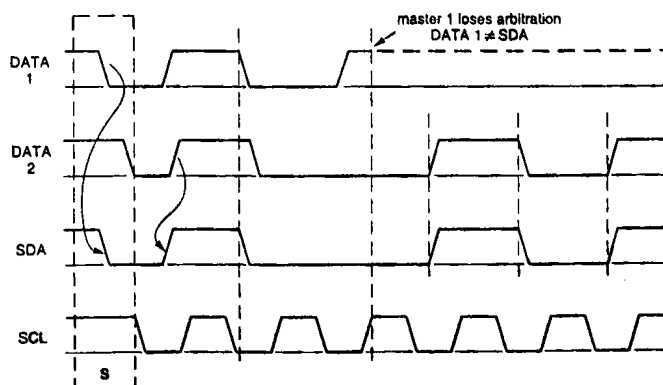
Σχήμα 1-6. Επιβεβαίωση στο δίαυλο I²C.

Όταν ένας slave δεν επιβεβαιώνει τη διεύθυνση Slave που έχει μεταδοθεί, πρέπει να κρατήσει τη γραμμή SDA σε υψηλή λογική κατάσταση. Ο master μπορεί, επομένως, να εκπέμψει μια διαδικασία τερματισμού (P) είτε μια επαναληπτική διαδικασία έναυσης (Sr).

Αν μια συσκευή slave-αποδέκτης επιβεβαιώσει τη διεύθυνση Slave, αλλά στη συνέχεια δεν μπορεί να λάβει περισσότερα bytes δεδομένων πρέπει να ενημερώσει τη συσκευή master κρατώντας τη γραμμή SDA σε υψηλή λογική κατάσταση. Στη συνέχεια, η συσκευή master εκπέμπει μια διαδικασία τερματισμού. Η περίπτωση αυτή ονομάζεται not-acknowledge (βλέπε σχήμα 1-6).

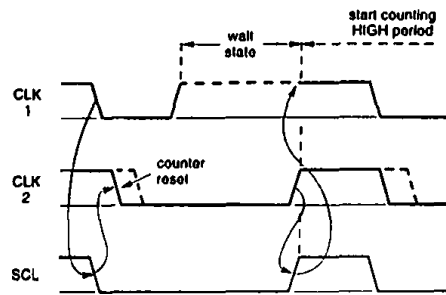
1.6 Arbitration

Στην περίπτωση που δύο ή περισσότεροι master προσπαθήσουν να αποστείλουν πληροφορίες στο δίαυλο, εκείνος που θα αποστείλει λογική κατάσταση "1" την ίδια χρονική στιγμή που κάποιος άλλος αποστέλλει λογικό "0" θα χάσει την κυριότητα του διαύλου (βλέπε σχήμα 1-7).



Σχήμα 1-7. Διαίτησία μεταξύ δύο συσκευών master.

Το SCL (serial clock) κατά τη διαδικασία της διαίτησίας είναι ένας συγχρονισμένος συνδυασμός των σημάτων συγχρονισμού που παράγουν οι master (βλέπε σχήμα 1-8).



Σχήμα 1-8. Συγχρονισμός παλμών ρολογιού κατά τη φάση διαίτησίας.

Η παραγωγή σημάτων συγχρονισμού στο δίαυλο I²C είναι πάντα αρμοδιότητα των συσκευών master, κάθε master παράγει τα δικά του σήματα clock όταν μεταφέρει πληροφορίες στο δίαυλο. Τα σήματα συγχρονισμού που παράγονται από ένα αφέντη μπορούν να μεταβληθούν μόνο όταν κάποια αργή συσκευή slave καθυστερεί τη γραμμή του ρολογιού ή κατά τη φάση διαίτησίας από κάποιον άλλο αφέντη.

1.7 Διευθυνσιοδότηση

Η διευθυνσιοδότηση για τον δίαυλο I²C γινόταν αρχικά με τη χρήση ενός byte (εξακολουθεί να χρησιμοποιείται και σήμερα). Το τελευταίο bit του byte διευθυνσιοδότησης δείχνει αν ο master επιθυμεί διαδικασία εγγραφής ή ανάγνωσης με τον απευθυνόμενο slave. Πρόκειται δηλαδή για διευθυνσιοδότηση 7 bits. Χρησιμοποιώντας 7 bits μπορούν να υπάρξουν μέχρι 127 διαφορετικές διευθύνσεις.

Στην πραγματικότητα, οι δυνατές διευθύνσεις είναι λιγότερες (112) γιατί υπάρχουν κάποιες διευθύνσεις δεσμευμένες τόσο για τη λειτουργία του διαύλου I²C όσο και για τη συμβατότητα με άλλους διαύλους. Στον πίνακα που ακολουθεί, υπάρχουν οι διευθύνσεις που είναι δεσμευμένες για το δίαυλο I²C

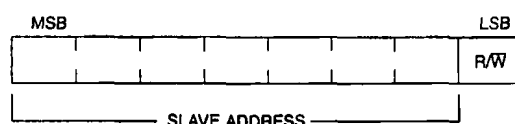
Διεύθυνση Slave	R/ \bar{W} bit	Περιγραφή
0000 000	0	Διεύθυνση General Call
0000 000	1	START byte (multimaster mode)
0000 001	X	Διεύθυνση CBUS (άλλος δίαυλος)
0000 010	X	Δεσμευμένη για μελλοντική χρήση
0000 011	X	Δεσμευμένη για μελλοντική χρήση
0000 1XX	X	High Speed master code
1111 1XX	X	Δεσμευμένη για μελλοντική χρήση
1111 0XX	X	Διευθυνσιοδότηση 10 bit

Πίνακας 1

Λόγω της απήχησης που είχε ο δίαυλος I²C και σε άλλες εταιρίες εκτός από τη Phillips, έπρεπε να βρεθεί ένας τρόπος για επέκταση των διευθύνσεων. Αυτό επιτεύχθηκε με τη χρήση 10 bits αντί 7 bits για τη διευθυνσιοδότηση. Παράλληλα, η επέκταση στα 10 bits αλλαγή έπρεπε να είναι συμβατή με τη διευθυνσιοδότηση των 7 bits.

1.7.1 Διευθυνσιοδότηση 7 bits

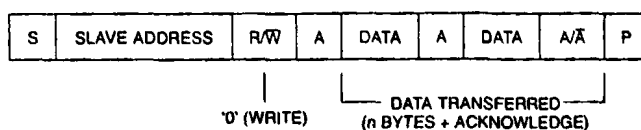
Κατά τη λειτουργία με διευθυνσιοδότηση 7 bits, μετά τη διαδικασία έναυσης START (S) αποστέλλεται στο δίαυλο η διεύθυνση slave. Η διεύθυνση αυτή αποτελείται από 7 bits, τα οποία ακολουθούνται από ένα όγδοο bit (για να συμπληρωθεί ένα byte). Το τελευταίο καθορίζει την κατεύθυνση των δεδομένων. Το «0» αντιστοιχεί στην εκπομπή (Εγγραφή προς το δίαυλο) (WRITE) και το «1» δηλώνει την αίτηση του master για ανάγνωση δεδομένων από το δίαυλο (READ) (βλέπε σχήμα 1-9). Μια διαδικασία μεταφοράς δεδομένων τερματίζεται πάντα από μια διαδικασία τερματισμού (STOP) (P) που εκπέμπεται από τον master.



Σχήμα 1-9. Το πρώτο Byte μετά τη διαδικασία START.

Οι πιθανές διαδικασίες μεταφοράς είναι οι ακόλουθες:

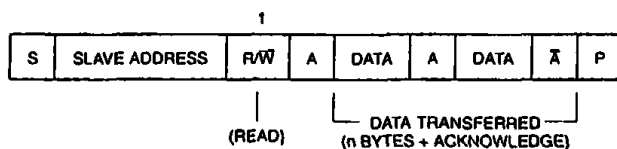
- Ο Master-transmitter αποστέλλει δεδομένα στον slave-receiver. Η φορά των δεδομένων δεν αλλάζει. (Μετά την αποστολή του πρώτου Byte ο master συνεχίζει με την αποστολή του δεύτερου) (σχήμα 1-10).
- Ο Master διαβάζει από τον slave αμέσως μετά την αποστολή του πρώτου byte (σχήμα 1-11).
- Συνδυασμένη διαδικασία (Σχήμα 1-12). Κατά την αλλαγή κατεύθυνσης των δεδομένων (μετά από εγγραφή ακολουθεί ανάγνωση ή αντίστροφα) σε μια μεταφορά, η διαδικασία START και η αποστολή της διεύθυνσης slave επαναλαμβάνονται και οι δύο, αλλά με το bit R/\bar{W} ανεστραμμένο. Εάν ο master receiver αποστείλει μια επαναληπτική διαδικασία START, πρέπει προηγουμένως να έχει στείλει ένα not-acknowledge \bar{A} .



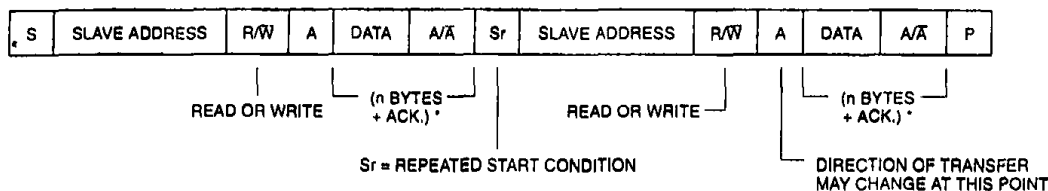
- FROM MASTER TO SLAVE
- FROM SLAVE TO MASTER

- A = ACKNOWLEDGE (SDA LOW)
- \bar{A} = NOT ACKNOWLEDGE (SDA HIGH)
- S = START CONDITION
- P = STOP CONDITION

Σχήμα 1-10. Ο master-πομπός απευθύνεται στον slave-αποδέκτη με διευθυνσιοδότηση 7 bits. Η κατεύθυνση της μετάδοσης δεν αλλάζει



Σχήμα 1-11. Ο master διαβάζει τον slave αμέσως μετά την αποστολή του πρώτου Byte.



Σχήμα 1-12. Συνδυασμένη λειτουργία.

1.7.2 Διευθυνσιοδότηση 10 bits

Η διεθυνσιοδότηση των 10 bits δεν αλλάζει τη δομή των δεδομένων του διαύλου I²C. Η διεθυνσιοδότηση των 10 bits χρησιμοποιεί τις δεσμευμένες διευθύνσεις της διεθυνσιοδότησης των 7 bits του πίνακα (1).

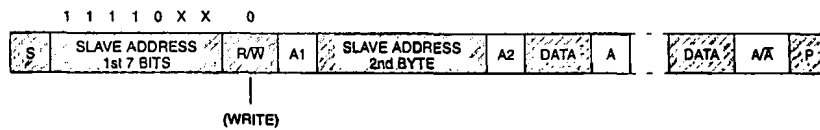
Η διεθυνσιοδότηση των 10 bits σχηματίζεται από τα 2 πρώτα bytes που ακολουθούν μια διαδικασία START (S) ή μια επαναληπτική διαδικασία START (Sr). Τα πρώτα 7 bits του πρώτου byte αποτελούν το συνδυασμό 11110XX του οποίου τα δύο τελευταία bits (XX) αποτελούν τα δύο περισσότερο σημαντικά bits (MSB) της διεύθυνσης των 10 bits, ενώ το όγδοο bit R/\bar{W} καθορίζει όπως προαναφέραμε την κατεύθυνση των δεδομένων (βλέπε σχήμα 1-13).

Αν το bit R/\bar{W} είναι '0', τότε το δεύτερο byte περιέχει τα υπόλοιπα 8 bits της διεύθυνσης των 10 bits. Αν το bit R/\bar{W} είναι '1' (ανάγνωση), τότε το byte που ακολουθεί περιέχει δεδομένα που αποστέλλονται από έναν slave προς έναν master.

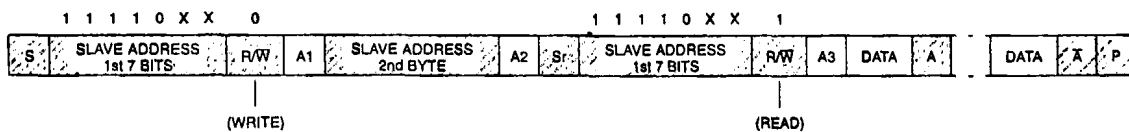
Οι πιθανές διαδικασίες μεταφοράς είναι οι ακόλουθες:

- Ο Master-transmitter αποστέλλει δεδομένα σε έναν slave-receiver με διεύθυνση των 10 bits. Η κατεύθυνση της μεταφοράς δεν αλλάζει. (Σχήμα 1-13). Μετά τη διεύθυνση των 8 bits θα απαντήσουν πιθανώς περισσότερες συσκευές slave, αλλά μετά το δεύτερο byte θα επιβεβαιώσει μια μόνο συσκευή (ACK 2)

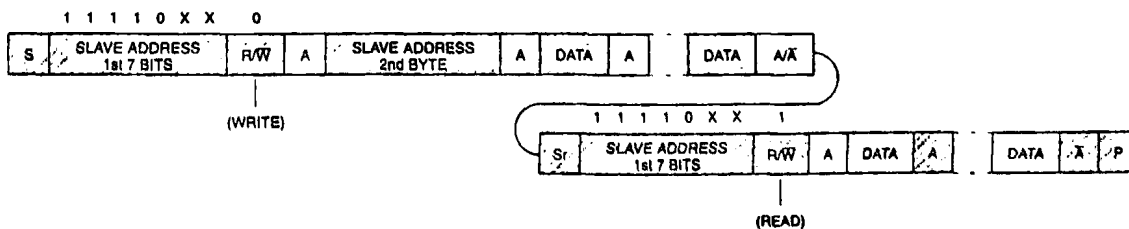
- Master-receiver διαβάζει τον slave-transmitter με διεύθυνση 10 bits. Η κατεύθυνση της μεταφοράς αλλάζει μετά το δεύτερο κατά σειρά R / \bar{W} bit (σχήμα 1-14). Όμοια με την προηγούμενη περίπτωση, μέχρι και το A2 ο συγκεκριμένος slave συνεχίζει να περιμένει εντολή από τον master. Μετά από μια επαναληπτική διαδικασία START, ο συγκεκριμένος slave αποστέλλει δεδομένα περιμένοντας μια διεύθυνση slave address που περιέχει μόνο τα 2 πρώτα bits της διευθυνσιοδότησης των 10 bits (σχήμα 1-15).
- Συνδυασμένη διαδικασία: Ο master αποστέλλει δεδομένα στον slave και κατόπιν λαμβάνει δεδομένα από τον ίδιο slave (σχήμα 1- 16).
- Συνδυασμένη διαδικασία: Διευθυνσιοδότηση 7 bits και 10 bits (σχήμα 1-17).



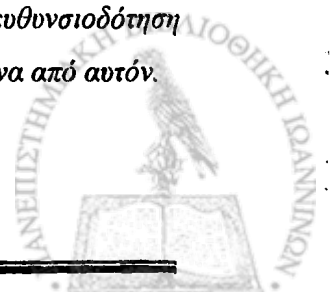
Σχήμα 1-13. Ο master-πομπός απευθύνεται στον slave-αποδέκτη με διευθυνσιοδότηση 10 bits.

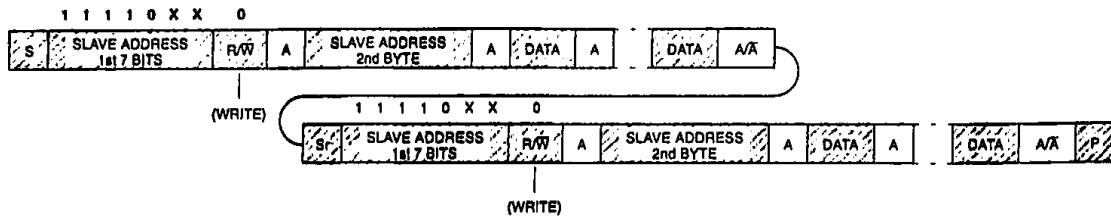


Σχήμα 1-14. Ο master-αποδέκτης απευθύνεται στον slave-πομπό με διευθυνσιοδότηση 10 bits.

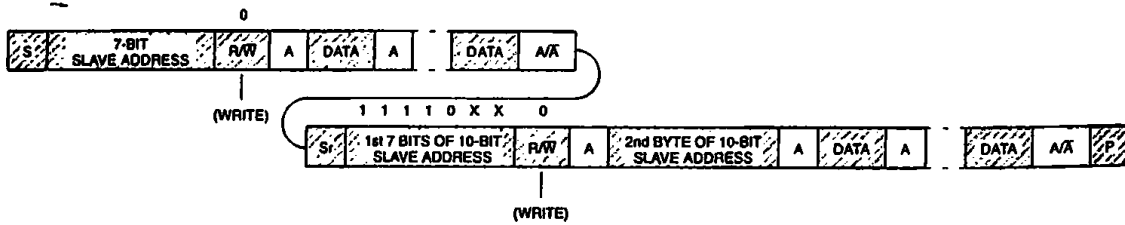


Σχήμα 1-15. Συνδυασμένη διαδικασία. Ο master απευθύνεται στον slave με διευθυνσιοδότηση 10 bits, στη συνέχεια αποστέλλει δεδομένα σε αυτόν και μετά λαμβάνει δεδομένα από αυτόν.





Σχήμα 1-16. Συνδυασμένη διαδικασία. Ο master αποστέλλει δεδομένα σε δυο slaves με διεθυνσιοδότηση 10 bits



Σχήμα 1-17. Συνδυασμένη διαδικασία. Ο master αποστέλλει δεδομένα σε δυο slaves. Στον έναν με διεθυνσιοδότηση 10 bits και στον άλλο με διεθυνσιοδότηση 7 bits.

2. Εισαγωγή στο PCI

2.1 Ο διάυλος PCI και η ιστορία των διαύλων στους προσωπικούς Η/Υ

Το 1985, η IBM κατασκεύασε τον πρώτο υπολογιστή AT. Οι θυγατρικές κάρτες του επικοινωνούσαν με τον επεξεργαστή, αξιοποιώντας τα σήματα του διαύλου ISA (INDUSTRY STANDARD ARCHITECTURE), που ήταν ήδη γνωστός από τους παλαιότερους επεξεργαστές της οικογένειας XT. Η χαμηλή διαμεταγωγή του ISA, επιβράδυνε σημαντικά τον υπολογιστή, κάνοντας τους επεξεργαστές x286 να φαίνονται ελάχιστα καλύτεροι από τους 8088 των XT. Η αδυναμία του διαύλου ISA έγινε ακόμα πιο φανερή, όταν έπρεπε να ανταλλάξει δεδομένα με γρήγορες (για την εποχή εκείνη) κάρτες γραφικών ή δικτύων. Η αιτία της βραδύτητας, βρισκόταν στην ίδια τη φύση του διαύλου, που για λόγους συμβατότητας λειτουργούσε σε συχνότητα 6 MHz. Η αύξηση της συχνότητας σε 8 MHz, βελτίωσε κάπως τον ρυθμό διαμεταγωγής, αλλά όχι τόσο, όσο έπρεπε. Η ταχύτητα έφθανε θεωρητικά τα 8 MB/sec με εύρος λέξης 16 bit. Στην πράξη όμως, τα πράγματα ήταν διαφορετικά. Η πραγματική ταχύτητα κυμαινόταν μεταξύ 4 και 6 MB/sec. [2]

➤ Ο διάυλος MCA

Η χαμηλή ταχύτητα του ISA, αποτέλεσε την αφορμή για μια καινούργια προσπάθεια της IBM με στόχο την υλοποίηση ενός ταχύτερου διαύλου. Ο διάυλος αυτός ονομάστηκε Micro Channel Architecture (MCA) και ήταν ικανός να μεταφέρει λέξεις των 32 bit (δεδομένα και διευθύνσεις). Η ταχύτητα μεταφοράς δεδομένων έφθανε τα 16 MB/sec. Οι υπολογιστές με διάυλο MCA, που έγιναν γνωστοί σαν PS/2, κατέληξαν σε εμπορική αποτυχία. Αιτία γι' αυτό ήταν η πλήρης απουσία των καταξιωμένων διαύλων ISA και κατ' επέκταση των καρτών αυτού του τύπου.

➤ Ο διάυλος EISA

Η αποτυχία της καθιέρωσης ενός τελειώς καινούργιου διαύλου, έκανε τους σχεδιαστές της IBM να αναθεωρήσουν τις απόψεις τους σχετικά με τους οποιουσδήποτε νέους διαύλους. Έτσι αποφάσισαν να βελτιώσουν, όσο ήταν δυνατόν, τον υπάρχοντα ISA. Ο διάυλος EISA, επέκταση του υπάρχοντος ISA, λειτουργούσε σε συχνότητες των 8 MHz και διακινούσε λέξεις εύρους 32 bit. Η ταχύτητα διαμεταγωγής του, έφθανε τα 16 MB/sec (κανονική λειτουργία) ή τα 32 MB/sec (burst mode). Λίγο αργότερα, ο διάυλος EISA αναβαθμίστηκε, επιτυγχάνοντας ρυθμούς μεταφοράς δεδομένων μέχρι και 133 MB/sec σε λειτουργία burst. Οι νέες αυτές παραλλαγές, έγιναν γνωστές με τους επιπλέον προσδιορισμούς EMB-66 και EMB-133. Οι μητρικές κάρτες που σχεδιάστηκαν για τις ανωτέρω εκδόσεις, όπως και οι ανάλογες θυγατρικές, κατέληξαν να φιλοξενούνται μόνο σε διακομιστές δικτύων, αφού το κόστος τους ήταν μεγάλο και τις καθιστούσε μη εμπορικές.

➤ διάυλος VLB

Οι ανάγκες της αγοράς, ώθησαν και πολλές άλλες εταιρίες να ασχοληθούν με τη σχεδίαση διαύλων. Το αποτέλεσμα ήταν η δημιουργία ενός καινούργιου διαύλου, ο οποίος χαρακτηριζόταν από χαμηλό κόστος και υψηλή απόδοση. Μετά από μερικές αποτυχημένες προσπάθειες, ο νέος διάυλος VLB (VESA Local Bus) φιλοξενήθηκε στις περισσότερες μητρικές της δεκαετίας του 1990. Υποστήριζε εύρος δεδομένων ίσο με 32 bits, με σήμα χρονισμού που ξεκινούσε από τα 25MHz και έφθανε τα 60 MHz (VLB-1.0 με σήμα χρονισμού 25-40 MHz, VLB-2.0 στα 25-50 MHz).

➤ Ο διάυλος PCI

Οι πρώτες σκέψεις για τη σχεδίαση του διαύλου Peripheral Component Interconnect (PCI), ξεκίνησαν στις αρχές του 1991. Η εταιρία που τον πρότεινε ήταν η Intel. Οι στόχοι που έπρεπε να πετύχει ο διάυλος ήταν: υψηλότερη ταχύτητα σε σχέση με αυτήν του ISA των 16 ψηφίων, καλύτερη ηλεκτρομαγνητική συμβατότητα (EMC) από αυτή των προκατόχων του, καθώς και η όσο το δυνατόν μακροβιότερη παρουσία του, στις μητρικές των υπολογιστών. Ένα χρόνο αργότερα, κυκλοφόρησαν οι πρώτες μητρικές και θυγατρικές κάρτες. Φυσικά ακολούθησαν αρκετές αναβαθμίσεις (1.0, 2.0, 2.1), που κατέληξαν στην έκδοση 2.2 που χρησιμοποιείται έως σήμερα. Κύριο γνώρισμα του διαύλου PCI, είναι η μεταφορά δεδομένων και διευθύνσεων εύρους 32 bit (ή 64 bit PCI V2.0) σε πολυπλεγμένη μορφή, μέσα από ένα σύνολο ισάριθμων γραμμών. Οι μεταφορές δεν είναι υποχρεωτικό να γίνονται πάντοτε στα 32/64 bit, αλλά μπορούν να προσαρμόζονται στις ανάγκες της εκάστοτε εφαρμογής. Το νέο αυτό είδος διαύλου, αποδείχθηκε το πλέον αποτελεσματικό για τους σύγχρονους προσωπικούς Η/Υ. Η πρώτη σειρά προδιαγραφών PCI (v1.0), αναπτύχθηκε όπως είδαμε από την Intel. Σήμερα την ανανέωση των προδιαγραφών PCI τη διαχειρίζεται μια ειδική ομάδα εταιριών η PCI SIG (Special Interest Group), στην οποία συμμετέχουν πολλές εταιρίες.

	ISA	EISA	Micro Channel	VL-Bus	PCI
Εύρος δεδομένων	8/16	32.00	16/32/64	32/64	32/64
Συχνότητα διαύλου (MHz)	5.33/8.33	8.33	10.00	33/50	33/66
Ρυθμός μεταφοράς δεδομένων (MB/sec)	5.33/8.33	33.00	20/40/80/160	132/264	132/264/528
Αριθμός υποδοχών	0-8	0-8	0-8	0-2	0-4
Υποστήριξη πολλών master	OXI	NAI	NAI	NAI	NAI
Ισοτιμία διευθύνσεων/δεδομένων	OXI	OXI	NAI	OXI	NAI
Sync, Channel Checks	OXI	OXI	NAI	OXI	NAI
Υποστήριξη αυτόματης εγκατάστασης/ρύθμισης	OXI	NAI	NAI	NAI	NAI

Πίνακας 2-1. Συγκριτικός πίνακας των βασικών χαρακτηριστικών των διαύλων στους Η.Υ.

2.2 Γενικά χαρακτηριστικά του PCI

PCI είναι το ακρωνύμιο για το Peripheral Component Interconnect και περιγράφει ένα δίαυλο υψηλών επιδόσεων, για τη διασύνδεση μεταξύ ολοκληρωμένων κυκλωμάτων, καρτών επέκτασης, επεξεργαστών και άλλων περιφερειακών για ηλεκτρονικούς υπολογιστές. Ο δίαυλος PCI, υποστηρίζει κάρτες που απαιτούν γρήγορη πρόσβαση είτε μεταξύ τους, είτε με τη μνήμη του συστήματος (ως σύστημα εννοείται ο H/Y στον οποίο φιλοξενούνται) και μπορεί να προσπελαστεί από τον επεξεργαστή σε ταχύτητες που πλησιάζουν την ονομαστική ταχύτητα του επεξεργαστή. Πρέπει να επισημανθεί ότι όλες οι συναλλαγές εγγραφής / ανάγνωσης στο δίαυλο PCI μπορούν να γίνουν τόσο σε χρόνο ενός κύκλου (single phases), όσο και κατά ριπές (burst mode transfer).

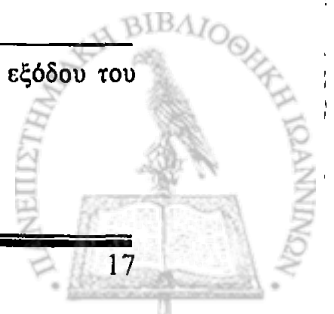
Ο δίαυλος PCI δεν αποτελεί ένα αληθινό Local Bus* αντιθέτως, καταλαμβάνει ένα ενδιάμεσο επίπεδο μεταξύ του διαύλου FSB, της κεντρικής μονάδας επεξεργασίας (CPU) και ενός κοινού διαύλου επέκτασης (πχ. ISA, Microchannel) (βλέπε σχήμα 2-1).

Ο δίαυλος PCI, απομονώνεται από το FSB με τη χρήση ενός ελεγκτή PCI (PCI Bridge). Ο ελεγκτής αυτός του διαύλου PCI, είναι γνωστός και με το όνομα North Bridge και είναι το βασικό ολοκληρωμένο κύκλωμα του συνόλου που υπάρχουν σε μια μητρική πλακέτα, τα οποία είναι γνωστά με τον όρο chipset. [3]

Η κεντρική μονάδα επεξεργασίας, μπορεί να αποστείλει δεδομένα στα περιφερειακά PCI, μέσω του εσωτερικού Buffer του North Bridge. Η διαδικασία αυτή, επιτρέπει στη CPU να ασχοληθεί με την επόμενη διεργασία, χωρίς να χρειάζεται να περιμένει το πέρας της μεταφοράς δεδομένων. Ο ελεγκτής North Bridge στη συνέχεια θα αποστείλει τα δεδομένα προς το περιφερειακό, με τον ταχύτερο δυνατό ρυθμό μετάδοσης.

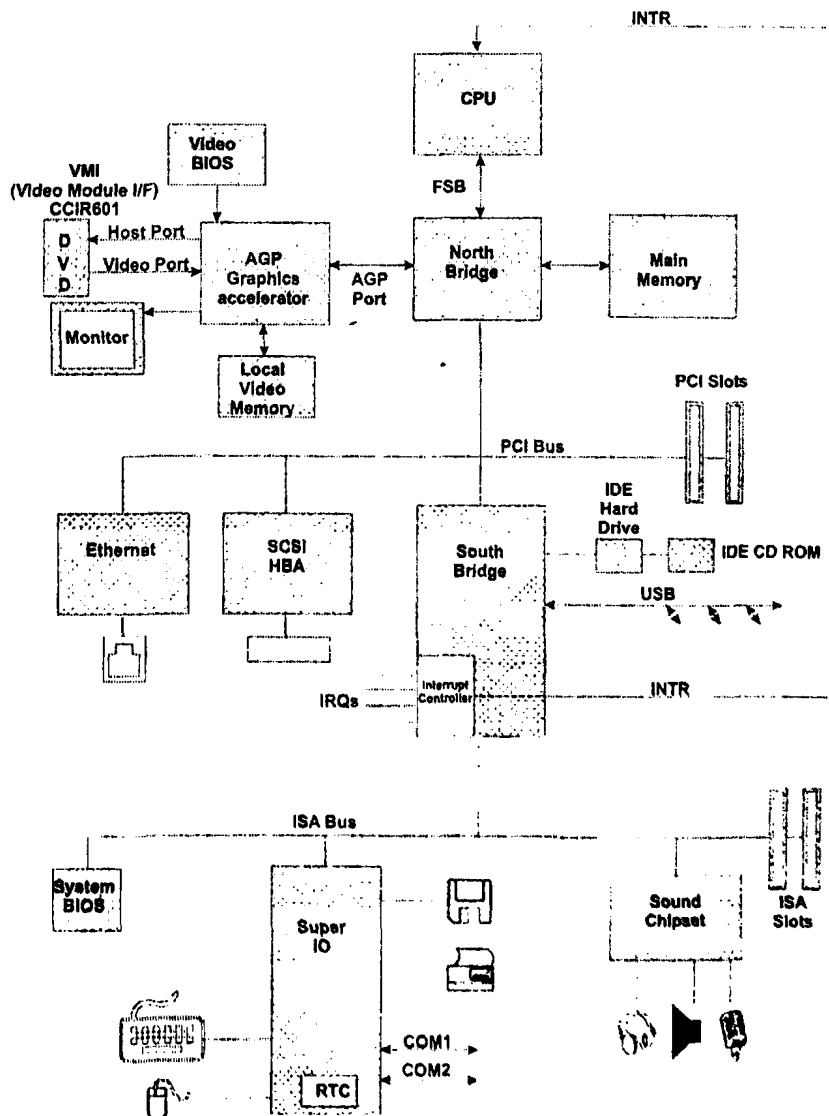
Η αρχιτεκτονική PCI υποστηρίζει επίσης πολλές συσκευές master οι οποίες μπορούν να αποκτήσουν την κυριότητα του διαύλου και να εκτελέσουν συναλλαγές στο δίαυλο, ανεξάρτητα από την κεντρική μονάδα επεξεργασίας.

* Με τον όρο Local Bus γινόταν αναφορά στο παρελθόν στο δίαυλο εισόδου / εξόδου του επεξεργαστή, σήμερα ο όρος Local Bus έχει αντικατασταθεί από τον όρο Front Side Bus



Το σχήμα 2-1 απεικονίζει τη βασική σχέση του διαύλου PCI, του FSB και των διαύλων επέκτασης. Συγκεκριμένα:

- Ο ελεγκτής North Bridge συνδέει το FSB με τον πρωτεύοντα δίαυλο PCI.
- Η γέφυρα PCI σε ISA, συχνά ονομαζόμενη και South Bridge, συνδέει τον πρωτεύοντα δίαυλο PCI με το δίαυλο ISA (ή EISA). Η γέφυρα South Bridge περιλαμβάνει επίσης τον ελεγκτή διακοπών (interrupt controller), τον ελεγκτή σκληρών δίσκων και περιφερειακών ATAPI, τη διεπαφή USB και τον ελεγκτή DMA όπως φαίνεται στο σχήμα 2-1.
- Μια ή περισσότερες γέφυρες PCI σε PCI μπορούν να υλοποιηθούν στον πρωτεύοντα δίαυλο PCI ή πάνω σε μια προσαρτώμενη κάρτα PCI.
- Το chipset, μπορεί να περιλαμβάνει περισσότερες του ενός North Bridges, για την υποστήριξη περισσότερων CPU.



Σχήμα 2-1. Τυπικό δομικό διάγραμμα ενός προσωπικού υπολογιστή

Πίνακας 2-2. Κυριότερα χαρακτηριστικά του PCI

Χαρακτηριστικά	Περιγραφή
Ο δίαυλος είναι ανεξάρτητος από το είδος του Επεξεργαστή	Τα εξαρτήματα που σχεδιάζονται, βασίζονται στον δίαυλο PCI και όχι στον επεξεργαστή. Έτσι, η σχεδίαση δεν χρειάζεται να αλλάξει σε πιθανή αλλαγή της αρχιτεκτονικής του επεξεργαστή.
Υποστήριξη για 80 περίπου λειτουργίες για κάθε δίαυλο PCI	Μια τυπική υλοποίηση PCI υποστηρίζει περίπου 10 φορτία (ηλεκτρικά) και κάθε συσκευή επιβαρύνει με ένα φορτίο τον δίαυλο. Κάθε συσκευή με την σειρά της, μπορεί να περιέχει μέχρι 8 λειτουργίες.
Υποστήριξη έως και 256 δίαυλων PCI	Οι προδιαγραφές του PCI προβλέπουν υποστήριξη για 256 δίαυλους PCI
Χαμηλή κατανάλωση ισχύος	Η σχεδίαση ενός συστήματος PCI γίνεται πάντα σύμφωνα με την προδιαγραφή PCI, κατά την οποία, μια συσκευή PCI πρέπει να απαιτεί για την λειτουργία της όσο το δυνατό μικρότερο ρεύμα.
Μεταφορές ριπής κατά την ανάγνωση και την εγγραφή	Ένας δίαυλος των 32-bit υποστηρίζει μεταφορά δεδομένων κατά ριπές με μέγιστο ρυθμό 132 Mbytes/Sec. Για ένα δίαυλο των 64-bit που λειτουργεί στα 33 MHz, ο ρυθμός φθάνει τα 264 Mbytes/Sec και για ένα δίαυλο των 64-bit που λειτουργεί στα 66 MHz, ο ρυθμός ανέρχεται στα 528 Mbytes/Sec.
Συχνότητα δίαυλου	Η έκδοση 2.0 του PCI υποστήριζε συχνότητα λειτουργίας του δίαυλου μέχρι και 33 MHz. Η έκδοση 2.1 του PCI προσθέτει υποστήριξη για συχνότητα λειτουργίας μέχρι και 66 MHz.
Εύρος διαύλου στα 64 bit	Πλήρης καθορισμός επέκτασης στα 64-bit, ενώ διατηρείται πάντα συμβατότητα με τα περιφερειακά PCI των 32-bit.
Χρόνος προσπέλασης	Μικρός μέχρι και 60nsec για συχνότητα λειτουργίας του δίαυλου στα 33 MHz κατά την μεταφορά από έναν initiator, σε ένα target.
Ταυτόχρονη λειτουργία δίαυλου	Οι γέφυρες υποστηρίζουν πλήρη, ταυτόχρονη λειτουργία με τον δίαυλο του επεξεργαστή και τους υπόλοιπους δίαυλους επέκτασης σε ένα σύστημα.
Υποστήριξη master	Η πλήρης υποστήριξη για πολλούς masters του δίαυλου, επιτρέπει επικοινωνία μεταξύ των συσκευών του ίδιου δίαυλου αλλά επίσης με την κεντρική μνήμη και τους δίαυλους επέκτασης. Επιπλέον ένας master PCI μπορεί να προσπελάσει μια συσκευή slave που φιλοξενείται σε κάποιον άλλο δίαυλο PCI χαμηλότερο στην ιεραρχία.
Μικρό πλήθος ακροδεκτών	Η οικονομική χρήση των σημάτων του δίαυλου, επιτρέπει την ύπαρξη ενός πλήρως λειτουργικού slave με μόλις 47 ακροδέκτες ενώ για έναν initiator χρειάζονται συνολικά 49 ακροδέκτες.
Έλεγχος εγκυρότητας μεταφοράς	Ο δίαυλος PCI υποστηρίζει έλεγχο ισοτιμίας σε επίπεδο διευθύνσεων, δεδομένων και εντολών.
Τρεις περιοχές μνήμης	Ο δίαυλος PCI υποστηρίζει τρεις περιοχές διευθύνσεων εισόδου/εξόδου, μνήμης και περιοχής ρυθμίσεων.
Αυτόματη Ρύθμιση	Υλοποίηση των καταχωρητών ρυθμίσεων σε επίπεδο bit για την υποστήριξη του αυτόματου εντοπιστού υλικού από το σύστημα και την εγκατάσταση αυτού.
Προσαρτώμενες περιφερειακές κάρτες	Οι προδιαγραφές PCI περιλαμβάνουν ορισμούς, τόσο για τους σύνδεσμους PCI, όσο και για προσαρτώμενες κάρτες.
Μέγεθος προσαρτώμενων καρτών	Οι προδιαγραφές PCI ορίζουν τρία μεγέθη για τις προσαρτώμενες κάρτες : μεγάλο, μικρό και μικρό μεταβλητού ύψους.

2.3 Βασικές έννοιες στο δίαυλο PCI

Όπως ήδη αναφέραμε, οι συσκευές που συναντώνται συνήθως σε ένα δίαυλο PCI, μπορούν να διαχωριστούν με μια πρώτη ταξινόμηση σε συσκευές master και slave. Συγκεκριμένα:

- Συσκευές **master** αποτελούν συνήθως «έξυπνες» συσκευές οι οποίες μπορούν να αποκτήσουν την κυριότητα του διαύλου.
- Συσκευές **slave** αποτελούν οι υπόλοιπες συσκευές του διαύλου που δεν μπορούν να αποκτήσουν την κυριότητα του και απαντούν στις επιταγές των συσκευών master.

Με μια δεύτερη ταξινόμηση, μπορούμε να διαχωρίσουμε τις συσκευές που υποστηρίζονται από το δίαυλο PCI σε Initiators και targets.

- **Initiators**, είναι οι συσκευές που ξεκινούν μια συναλλαγή στο δίαυλο PCI
- **Targets**, είναι οι συσκευές στις οποίες απευθύνονται οι initiators κατά την τρέχουσα συναλλαγή.

Από τα παραπάνω, φαίνεται, ότι initiators μπορούν να είναι μόνο συσκευές master, ενώ targets, μπορούν να είναι τόσο οι συσκευές slave, όσο και οι συσκευές master σε ένα δίαυλο PCI που φιλοξενεί πολλές συσκευές master.

Μια συναλλαγή στο δίαυλο PCI, λαμβάνει χώρα μεταξύ ενός Initiator και ενός target. Η συναλλαγή περιλαμβάνει αρχικά ένα στάδιο διευθυνσιοδότησης (Address phase), το οποίο ακολουθείται από ένα ή περισσότερα στάδια δεδομένων (data phases). Η συναλλαγή με πολλαπλά στάδια δεδομένων, ονομάζεται μετάδοση κατά ριπές (burst transfer). Το μήκος της μετάδοσης αυτής, καθορίζεται από τον εκάστοτε master. Η συσκευή target ενημερώνεται για τη διεύθυνση εκκίνησης, τον τύπο της συναλλαγής (εγγραφή ή ανάγνωση), τη χρονική στιγμή της εκκίνησης, αλλά όχι για το μήκος της συναλλαγής. Καθώς η συσκευή master ετοιμάζεται για τη μεταφορά του κάθε πακέτου δεδομένων, πληροφορεί τη συσκευή slave, αν πρόκειται για το τελευταίο πακέτο ή όχι. Η συναλλαγή ολοκληρώνεται με τη μεταφορά και του τελευταίου πακέτου δεδομένων.



2.4 Συσκευές PCI μιας ή πολλών λειτουργιών.

Ένα πραγματικό περιφερειακό PCI (physical PCI device) μπορεί να έχει την μορφή ενός εξαρτήματος υλοποιημένου στην μητρική πλακέτα του συστήματος, ή μπορεί να υλοποιηθεί με τη μορφή μιας προσαρτώμενης κάρτας για το δίαυλο PCI. Κάθε περιφερειακό PCI (ή αλλιώς συσκευή PCI) μπορεί να υλοποιήσει με τη σειρά του, από μια έως 8 ξεχωριστές λειτουργίες. Κάθε λειτουργία αποτελεί λογική συσκευή (logical device) για το σύστημα. [4]

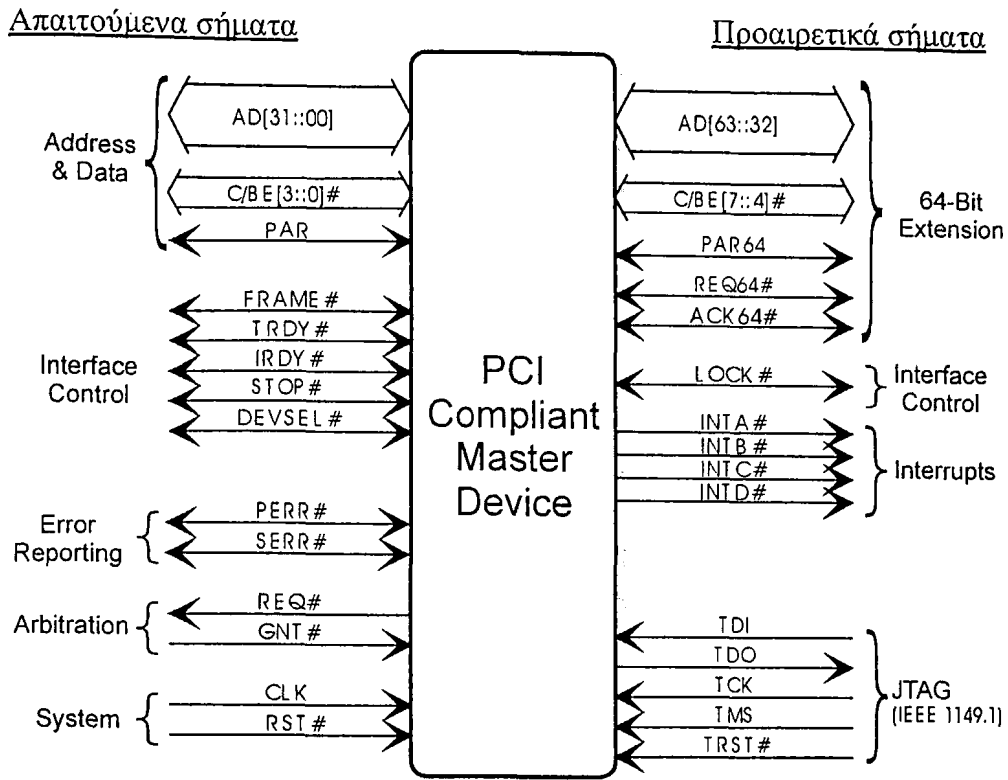
- Ένα περιφερειακό PCI που υλοποιεί μια μόνο λειτουργία, ονομάζεται **συσκευή PCI μιας λειτουργίας**,
- Ενώ ένα περιφερειακό που περιέχει 2 ή περισσότερες λειτουργίες καλείται **συσκευή PCI πολλών λειτουργιών**

Κάθε λειτουργία έχει τη δική της περιοχή ρυθμίσεων μεγέθους 256 bytes. Στην περιοχή αυτή, υλοποιούνται οι καταχωρητές ρυθμίσεων της λειτουργίας. Χρησιμοποιώντας τους καταχωρητές αυτούς το λογισμικό ρύθμισης (το BIOS του Η/Υ ή το λογισμικό οδήγησης) μπορεί να εντοπίσει αυτόματα την ύπαρξη της συσκευής (πραγματική ή λογική), να καθορίσει τις απαιτήσεις της σε πόρους συστήματος (περιοχή εισόδου / εξόδου, μνήμης, γραμμή διακοπής IRQ κ.α.) και κατόπιν να αναθέσει πόρους στη συσκευή οι οποίοι δεν έρχονται σε διένεξη με τους δεσμευμένους πόρους κάποιας άλλης συσκευής.

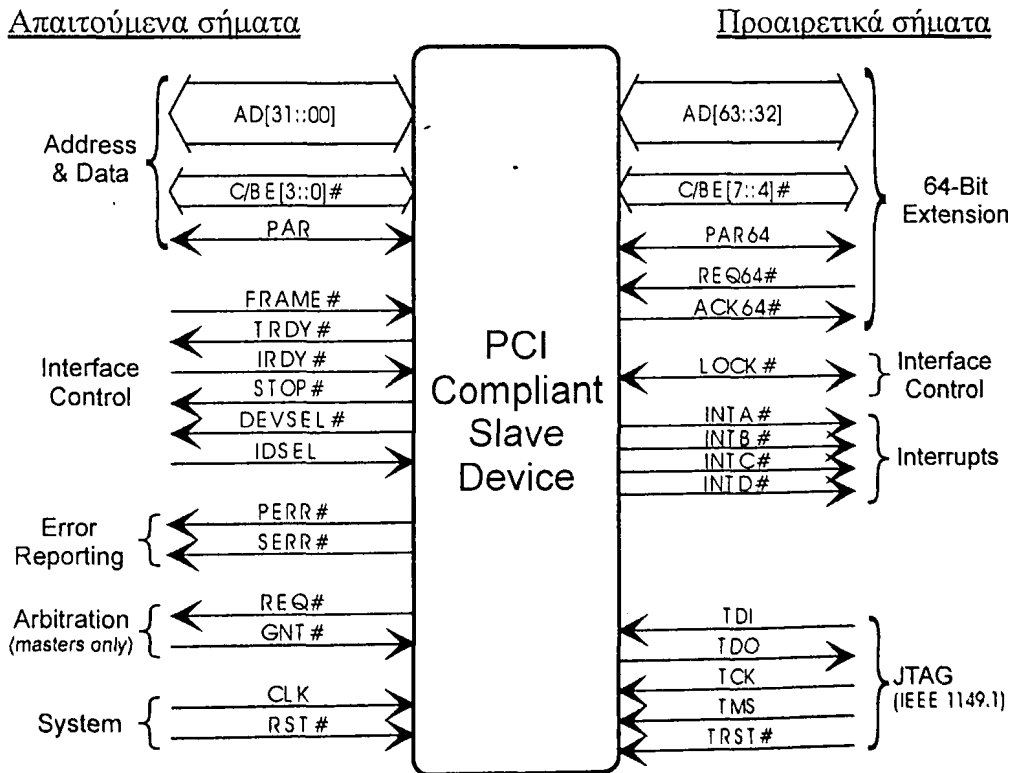
2.5 Οι ομάδες σημάτων του δίαυλου PCI

Σε αυτή την παράγραφο παραθέτονται τα σήματα που χρησιμοποιούνται για τη διασύνδεση μιας συσκευής, σύμφωνης με τις προδιαγραφές PCI, με το δίαυλο PCI. Στα σχήματα 2 και 3 που ακολουθούν εικονίζονται τα απαιτούμενα και προαιρετικά σήματα για συσκευές master και slave αντίστοιχα. Μια συσκευή PCI που ενεργεί άλλοτε ως initiator και άλλοτε ως target, πρέπει προφανώς να διαθέτει όλα τα σχετιζόμενα και με τους δυο τύπους συσκευών σήματα. Στην πραγματικότητα δεν υπάρχει συσκευή master που να μην είναι ποτέ target.

Κάθε ομάδα σημάτων περιγράφεται στο παράρτημα Δ. Πρέπει να σημειωθεί ότι κάποια από τα προαιρετικά σήματα δεν είναι προαιρετικά για κάποιους τύπους συσκευών PCI.



Σχήμα 2-2. Τα σήματα μιας master συσκευής PCI.

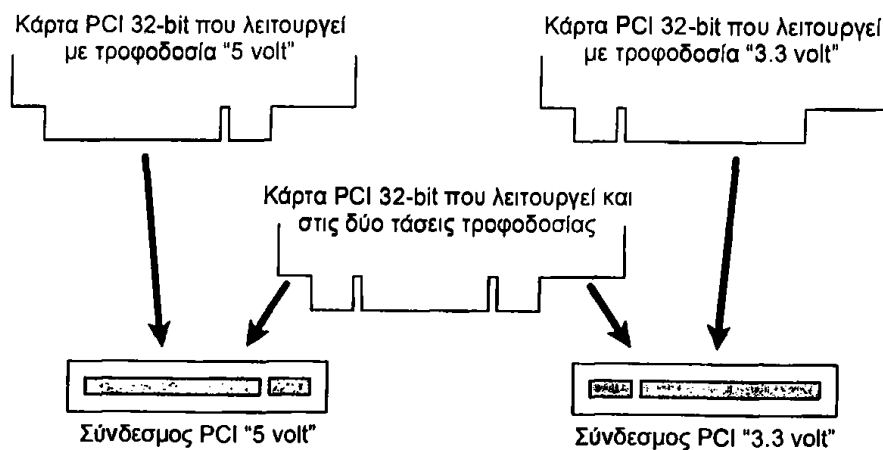


Σχήμα 2-3. Τα σήματα μιας slave συσκευής PCI.

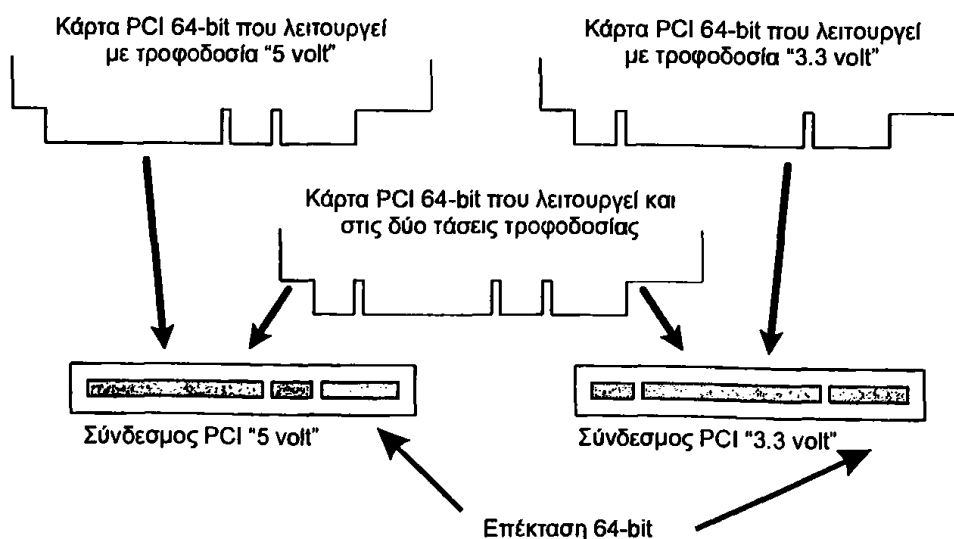


2.6 Τύποι προσαρτώμενων καρτών PCI

Η αρχιτεκτονική PCI ορίζει 2 τύπους συνδέσμων, 3 τύπους προσαρτώμενων καρτών καθώς και μια επέκταση συνδέσμου για τις κάρτες των 64 bit. Οι σύνδεσμοι διακρίνονται σύμφωνα με την τάση τροφοδοσίας που παρέχουν (5 Volt ή 3.3 Volt). Οι προσαρτώμενες κάρτες διακρίνονται σε τρεις. Μια για κάθε τύπο συνδέσμου (5 Volt ή 3.3 Volt) και μια τρίτη η οποία μπορεί να λειτουργήσει και στις 2 τάσεις τροφοδοσίας. Τέλος, όσον αφορά τις κάρτες των 64 bit, έχει προβλεφθεί μία επέκταση του συνδέσμου η οποία περιλαμβάνει τους επιπλέον ακροδέκτες για μεταφορές δεδομένων στα 64-bit. Αξίζει να σημειωθεί, ότι αν και η έκδοση PCI V2.2. εκδόθηκε στις αρχές του 1999 η σύνδεσμοι PCI 64-bit εξακολουθούν να εμφανίζονται μόνο σε ελάχιστες μητρικές κάρτες. Οι σύνδεσμοι 64-bit είναι πλήρως συμβατοί με τις κάρτες 32-bit.



Σχήμα 2-4. Κάρτες και σύνδεσμοι PCI 32-bit.

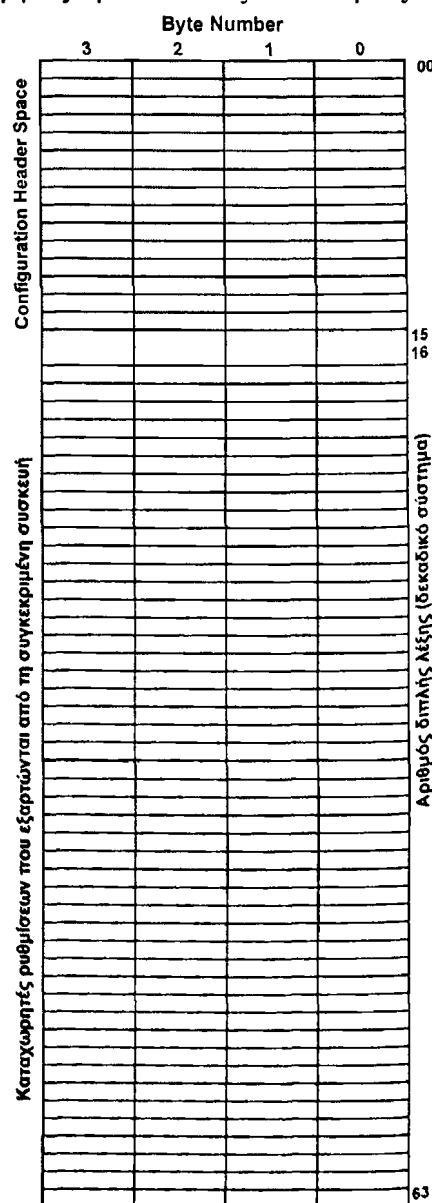


Σχήμα 2-5. Κάρτες και σύνδεσμοι PCI 64-bit.

2.7 Περιοχή ρυθμίσεων

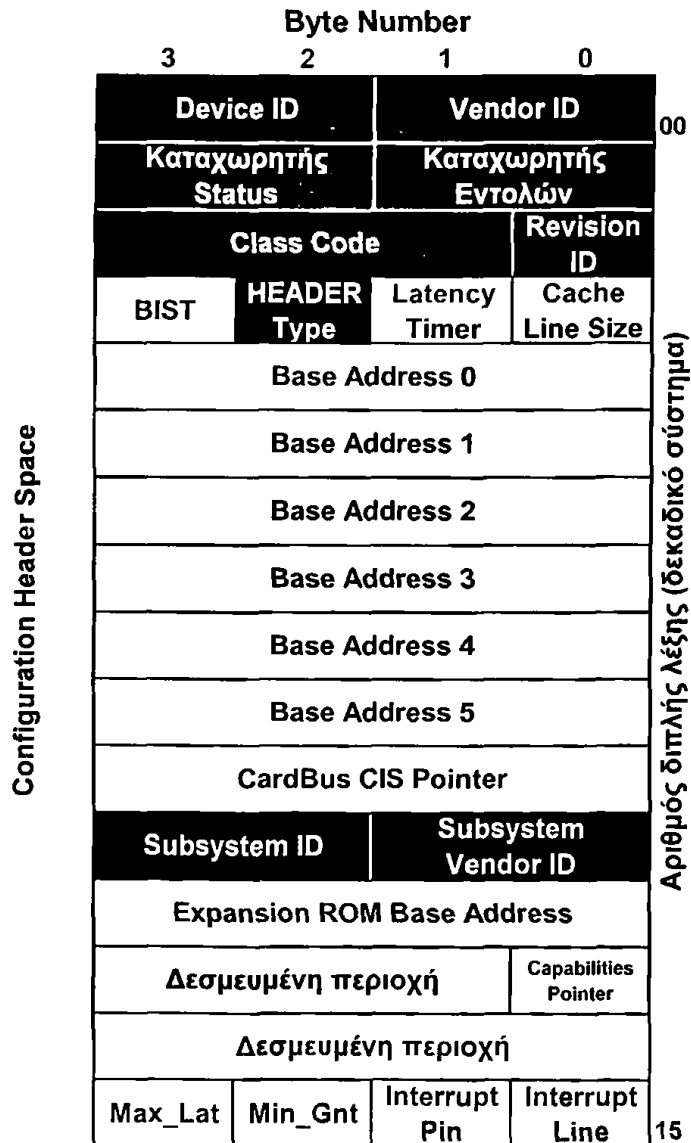
Αμέσως μετά την εκκίνηση του υπολογιστή, το BIOS (Basic Input Output System) εντοπίζει τις κάρτες PCI και ενημερώνεται από αυτές για τους πόρους που επιθυμούν να δεσμεύσουν. Τις περισσότερες φορές, οι ζητούμενοι πόροι αφορούν σήματα διακοπής, διευθύνσεις εισόδου / εξόδου, κανάλια απευθείας πρόσβασης στη μνήμη (DMA) και γενικότερα περιοχές της κεντρικής μνήμης RAM του υπολογιστή. Αν κάποιος από τους πόρους που ζητά μια κάρτα PCI, συμπίπτει με πόρους που έχει δεσμεύσει κάποια άλλη, τότε το BIOS προσπαθεί να αναδρομολογήσει τους πόρους, έτσι ώστε να εξαλειφθούν οι διενέξεις. Μόνο στην ακραία περίπτωση που το BIOS αποτύχει στην προσπάθειά του, τίθεται εκτός λειτουργίας μία από τις δύο κάρτες. Στη συνέχεια, δημιουργείται ένας κατάλογος με τους δεσμευμένους πόρους (ESCD database), με σκοπό τη μεταφορά των πληροφοριών αυτών στο λειτουργικό σύστημα για την περαιτέρω αξιοποίηση τους από το λογισμικό οδήγησης. [5]

Το λογισμικό οδήγησης μπορεί να αλλάξει εκ των υστέρων τις αρχικές ρυθμίσεις του BIOS αλλά δεν υπάρχει λόγος να γίνει κάτι τέτοιο. Οι ίδιες πληροφορίες αποθηκεύονται επίσης και στην περιοχή ρυθμίσεων (Configuration Space) των καρτών PCI. Η περιοχή αυτή, είναι στην πραγματικότητα μια μνήμη χωρητικότητας 256 byte και οργανώνεται σε 2 μέρη, όπως φαίνεται στο διπλανό σχήμα. Η πρώτη περιοχή (Configuration Header Space), που εκτείνεται σε 16 διπλές λέξεις, είναι αυστηρά καθορισμένη για όλες τις συσκευές PCI ενώ η δεύτερη περιοχή σχετίζεται με τη λειτουργία της συσκευής. Οι συσκευές PCI υλοποιούν μόνο τους απαραίτητους και σχετικούς με τη λειτουργίας τους καταχωρητές σε κάθε μια από τις περιοχές αυτές.



Σχήμα 2-6. Ο χάρτης διευθύνσεων της περιοχής ρυθμίσεων για μια συσκευή PCI.

Όπως προαναφέραμε, τα πρώτα 64 byte τις περιοχής ρυθμίσεων, είναι αυστηρά καθορισμένα για όλες τις συσκευές PCI και έχουν τη διάταξη που εικονίζεται στο σχήμα 2-7. Οι καταχωρητές με μαύρο φόντο υλοποιούνται υποχρεωτικά για όλες τις συσκευές PCI ενώ οι υπόλοιποι καταχωρητές υλοποιούνται προαιρετικά. Η περιγραφή όλων των στοιχείων που δηλώνονται στη περιοχή ρυθμίσεων, ξεφεύγει από το πλαίσιο αυτής της εργασίας. Ωστόσο, γίνεται αναφορά στα σημεία που κρίνονται απαραίτητα.



Σχήμα 2-7. Ο χάρτης διευθύνσεων του Configuration Header μιας συσκευής PCI.

Τα πεδία Vendor ID, Device ID, Class, Subsystem ID και Subsystem Vendor ID φέρουν την ταυτότητα της συσκευής. Ενώ οι καταχωρητές Base Address 1 έως 5 φέρουν πληροφορίες για τις απαιτήσεις της συσκευής σε περιοχές εισόδου / εξόδου και μνήμης.

□ **Vendor ID**

Πρόκειται για ένα καταχωρητή 16-bit ο οποίος δηλώνει τον κατασκευαστή του περιφερειακού. Οι κωδικοί αριθμοί των εταιριών που δηλώνονται σε αυτό το πεδίο, διαχειρίζονται από τον οργανισμό PCISIG και είναι μοναδικοί για κάθε κατασκευαστή. Ως παράδειγμα μπορούμε να αναφέρουμε τον κωδικό αριθμό της Intel “8086” και της AMCC “10E8”.

□ **Device ID**

Πρόκειται για ένα καταχωρητή 16-bit, που περιέχει συμπληρωματικές πληροφορίες για το περιφερειακό τις οποίες διαχειρίζεται ο κατασκευαστής του.

□ **Class**

Τα περιφερειακά PCI ανήκουν σε κατηγορίες (classes) όπως έχουν οριστεί από το PCISIG. Ο αντίστοιχος καταχωρητής έχει εύρος 16-bit, εκ των οποίων τα πρώτα 8 bit δηλώνουν τη γενική κατηγορία (base class ή group). Έτσι, τα περιφερειακά Ethernet και Token ring ανήκουν στην γενική κατηγορία των περιφερειακών δικτύου.

□ **Subsystem ID και Subsystem Vendor ID**

Τα πεδία αυτά χρησιμοποιούνται για λεπτομερέστερη αναγνώριση των περιφερειακών PCI, τα οποία υλοποιούνται συνήθως στις μητρικές κάρτες των ηλεκτρονικών υπολογιστών και αντιμετωπίζονται διαφορετικά από το σύστημα.

□ **Base Address Registers**

Οι Base Address Registers (ή αλλιώς BARs), αποτελούν μια άλλη πολύ σημαντική κατηγορία καταχωρητών που σχετίζεται με τις απαιτήσεις του κάθε περιφερειακού σε περιοχή μνήμης και I/O. Αν και η υλοποίηση τους είναι προαιρετική, χρησιμοποιείται σχεδόν πάντα τουλάχιστον ένας από αυτούς ο BAR0. Στους καταχωρητές αυτούς, δηλώνεται το μέγεθος, των περιοχών μνήμης ή I/O που μπορεί να απαιτείται από το περιφερειακό. Η κατανομή των απαραίτητων πόρων του συστήματος επιτυγχάνεται με την αλληλεπίδραση των BARs με το BIOS.



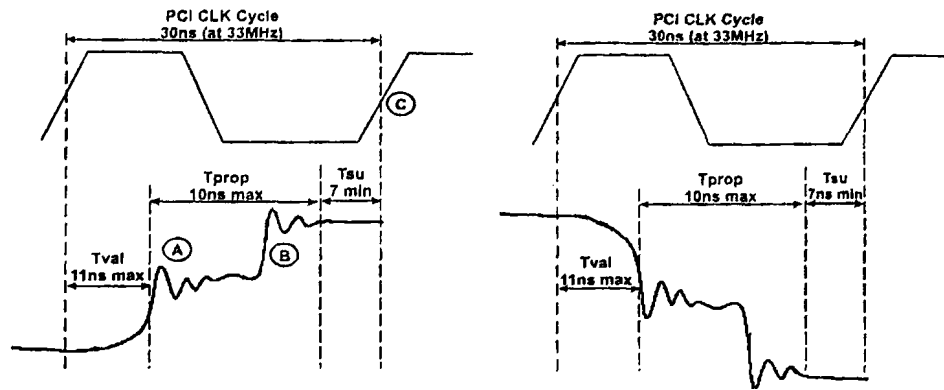
2.8 Μεταγωγή Ανακλώμενου Σήματος

Ο διάυλος PCI δεν τερματίζεται και χρησιμοποιεί την ανάκλαση σήματος. Μια προσεκτικά επιλεγμένη, αλλά σχετικά αδύνατη βαθμίδα οδήγησης χρησιμοποιείται για να οδηγήσει μερικώς τη γραμμή σήματος στην επιθυμητή λογική κατάσταση (βλέπε σχήμα 2-8). Η βαθμίδα οδήγησης πρέπει να οδηγεί μερικώς το σήμα προς την τελική κατάσταση και όχι τελείως (όπως θα γινόταν με μια ισχυρή βαθμίδα οδήγησης και με μια τερματισμένη γραμμή).

Όταν το μέτωπο κύματος φθάσει στο μη τερματισμένο άκρο του διαύλου, ανακλάται προς τα πίσω και διπλασιάζεται (σημείο B του σχήματος 3-2). Κατά την επιστροφή του σήματος κάθε είσοδος βλέπει ένα σωστό λογικό επίπεδο. Το σήμα δειγματοληπτείται από τους αποδέκτες με την έλευση του θετικού μετώπου του επόμενου παλμού ρολογιού (σημείο C, σχήμα 2-8). Τέλος, το μέτωπο του κύματος απορροφάται από την χαμηλή αντίσταση στο εσωτερικό της βαθμίδας οδήγησης. Αυτή η μέθοδος μειώνει το μέγεθος της βαθμίδας οδήγησης και το ρεύμα (surge) στο μισό.

Υπάρχουν τρεις χρονικές παράμετροι που σχετίζονται με τη διαδικασία:

- T_{val} : οι συσκευές PCI ξεκινούν την οδήγηση του σήματος στο θετικό μέτωπο του παλμού clock. T_{val} είναι το χρονικό διάστημα που χρειάζεται η βαθμίδα εξόδου να οδηγήσει το σήμα κατά ένα βήμα προς την τελική κατάσταση. Η βαθμίδα εξόδου πρέπει να φροντίσει να έχει τέτοια τιμή η τάση εξόδου, ώστε στη συνέχεια να αποκτήσει την επιθυμητή στάθμη.
- T_{prop} (propagation delay): Είναι το χρονικό διάστημα που χρειάζεται το wavefront για να ταξιδέψει μέχρι το άλλο άκρο του αγωγού να ανακλαστεί και να επιστρέψει.
- T_{su} (setup time): Είναι ο ελάχιστος χρόνος για τον οποίο το σήμα βρίσκεται στην τελική του κατάσταση πριν την έλευσή του θετικού μετώπου του επόμενου παλμού ρολογιού.
- T_h (hold time): Είναι ο χρόνος που πρέπει να παραμένουν τα σήματα στην τρέχουσα λογική κατάσταση μετά τη δειγματοληψία. Ο χρόνος αυτός για τα σήματα PCI είναι πρακτικά μηδέν.



Σχήμα 2-8. Κατά τη μετάβαση από '0' σε '1' και αντίστροφα το σήμα ανακλάται και διπλασιάζεται.

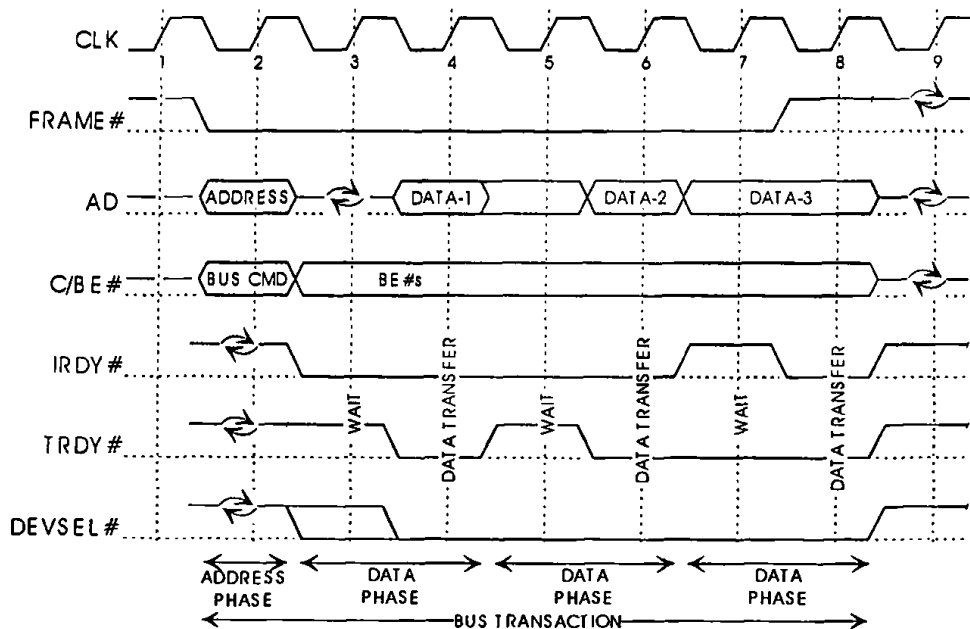
2.9 Χρονικά διαγράμματα του διαύλου PCI

Στη συνέχεια δίνονται δύο παραδείγματα μεταφοράς δεδομένων, ένα για ανάγνωση και ένα για εγγραφή (σχήματα 2-9 και 2-10 αντίστοιχα).

Μια διαδικασία ανάγνωσης περιλαμβάνει τα εξής στάδια :

- Cycle 1 → Ο δίαυλος είναι διαθέσιμος.
- Cycle 2 → Ο initiator παρέχει μια έγκυρη διεύθυνση και οδηγεί τα σήματα C/BE# για ανάγνωση. Αυτό είναι το στάδιο διευθυνσιοδότησης.
- Cycle 3 → Ο initiator απελευθερώνει τις γραμμές AD για να μπορέσει ο target να στείλει τα δεδομένα ανάγνωσης. Ο initiator οδηγεί τα σήματα C/BE# με έγκυρα δεδομένα για την ενεργοποίηση των. Ο initiator θέτει το IRDY# σε χαμηλό λογικό επίπεδο δείχνοντας έτσι ότι είναι έτοιμος να καταγράψει τα δεδομένα της ανάγνωσης. Ο target θέτει το DEVSEL# σε '0' (στον τρέχοντα ή στο επόμενο κύκλο) για να επιβεβαιώσει ότι έχει αποκωδικοποιήσει σωστά την δοθείσα διεύθυνση. Ο target οδηγεί το TRDY# σε λογικό '1' δείχνοντας ότι δεν έχει έτοιμα ακόμη τα δεδομένα προς αποστολή.
- Cycle 4 → Ο target έχει πλέον έτοιμα τα δεδομένα και θέτει το TRDY# σε λογικό '0' δείχνοντας στον initiator ότι τα δεδομένα είναι πλέον έγκυρα. Τα σήματα IRDY# και TRDY# παραμένουν και τα δύο σε χαμηλό επίπεδο και έτσι λαμβάνει χώρα η μεταφορά. Ο initiator καταγράφει (αποθηκεύσει) τα δεδομένα. Αυτό είναι το πρώτο στάδιο δεδομένων.
- Cycle 5 → Ο target απελευθερώνει το TRDY# (λογικό '1') δείχνοντας ότι χρειάζεται περισσότερο χρόνο να ετοιμαστεί για την επόμενη μεταφορά δεδομένων.
- Cycle 6 → Το δεύτερο στάδιο δεδομένων λαμβάνει χώρα όταν και τα δύο σήματα IRDY# και TRDY# είναι σε λογικό '0'. Ο initiator καταγράφει τα δεδομένα που παρέχονται από τον target.

- Cycle 7 → Ο target είναι έτοιμος για το τρίτο στάδιο δεδομένων αλλά ο initiator γνωστοποιεί ότι δεν είναι έτοιμος θέτοντας το IRDY# σε λογικό '1'.
- Cycle 8 → Ο initiator επανατοποθετεί το IRDY# σε λογικό '0' για να ολοκληρωθεί και το τρίτο στάδιο δεδομένων. Ο initiator καταγράφει τα δεδομένα που παρέχονται από τον target. Ο initiator οδηγεί το σε λογικό '1' δείχνοντας ότι πρόκειται για το τελευταίο στάδιο δεδομένων (τερματισμό από τον master).
- Cycle 9 → Οι γραμμές FRAME#, AD, και C/BE# απελευθερώνονται, ενώ οι γραμμές IRDY#, TRDY#, και DEVSEL# οδηγούνται σε λογικό '1' (απενεργοποίηση) για ένα κύκλο πριν απελευθερωθούν.

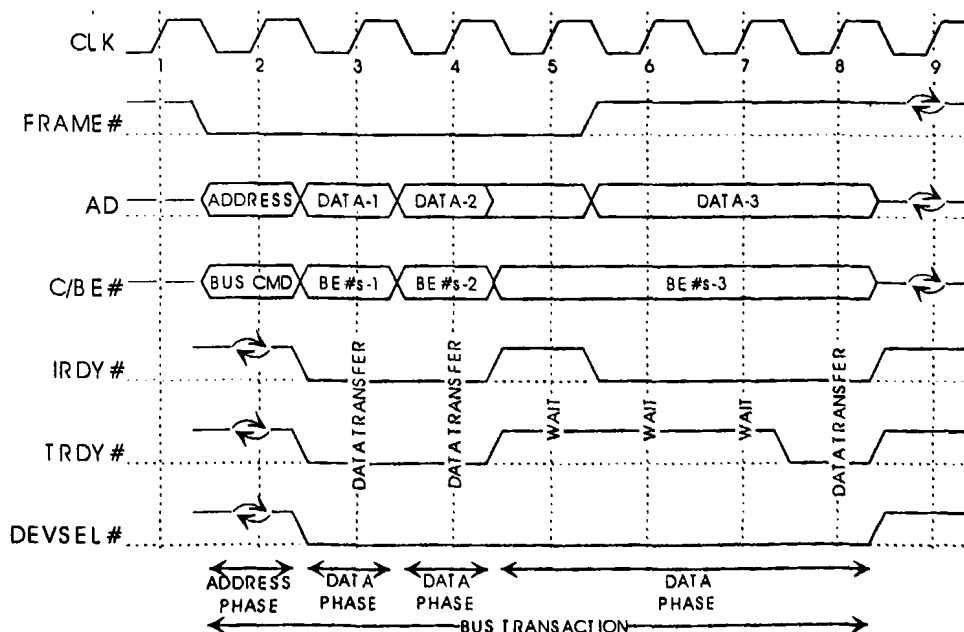


Σχήμα 2-9. Ο χρονικός συσχετισμός των σημάτων κατά την ανάγνωση.

Μια διαδικασία εγγραφής περιλαμβάνει τα εξής στάδια :

- Cycle 1 → Ο διάυλος είναι διαθέσιμος..
- Cycle 2 → Ο initiator αποστέλλει μια έγκυρη διεύθυνση και οδηγεί τα σήματα C/BE# για εγγραφή. Αυτό είναι το στάδιο διευθυνσιοδότησης.
- Cycle 3 → Ο initiator οδηγεί έγκυρα δεδομένα προς εγγραφή καθώς και σήματα ενεργοποίησης BE#. Ο initiator θέτει το IRDY# σε λογικό '0' δείχνοντας ότι είναι διαθέσιμα έγκυρα δεδομένα προς εγγραφή. Ο target θέτει το DEVSEL# σε λογικό '0' για να επιβεβαιώσει ότι έχει αποκωδικοποιήσει σωστά την δοθείσα διεύθυνση. Ο target θέτει το TRDY# σε λογικό '0' δείχνοντας ότι είναι έτοιμος να καταγράψει τα δεδομένα. Το πρώτο στάδιο δεδομένων λαμβάνει χώρα καθώς κατά τα δύο σήματα IRDY# και TRDY# είναι σε λογικό '0'. Ο target καταγράφει τα δεδομένα εγγραφής.

- Cycle 4→ Ο initiator παρέχει νέα δεδομένα και σήματα ενεργοποίησης BE#. Το δεύτερο στάδιο δεδομένων λαμβάνει χώρα καθώς κατά τα δύο σήματα IRDY# και TRDY# είναι σε λογικό '0'. Ο target καταγράφει τα δεδομένα εγγραφής.
- Cycle 5→ Ο initiator θέτει το IRDY#, γνωστοποιώντας ότι δεν είναι έτοιμος για το επόμενο στάδιο δεδομένων. Ο target θέτει το TRDY# γνωστοποιώντας ότι δεν είναι έτοιμος να δεχθεί δεδομένα.
- Cycle 6→ Ο initiator παρέχει έγκυρα δεδομένα για την επόμενη μεταφορά και θέτει το IRDY# σε λογικό '0'. Ο initiator οδηγεί το FRAME# σε λογικό '1' δείχνοντας ότι πρόκειται για το τελευταίο στάδιο δεδομένων (τερματισμός από τον master). Ο target παραμένει ανέτοιμος και κρατά τη γραμμή TRDY# σε λογικό '1'.
- Cycle 7→ Ο target παραμένει ανέτοιμος και κρατά τη γραμμή TRDY# σε λογικό '1'.
- Cycle 8→ Ο target είναι έτοιμος και θέτει τη γραμμή TRDY# σε λογικό '0'. Το τρίτο στάδιο δεδομένων, λαμβάνει χώρα καθώς κατά τα δύο σήματα IRDY# και TRDY# είναι σε λογικό '0'. Ο target καταγράφει τα δεδομένα εγγραφής.
- Cycle 9→ Οι γραμμές FRAME#, AD, και C/BE# απελευθερώνονται, ενώ οι γραμμές IRDY#, TRDY#, και DEVSEL# οδηγούνται σε λογικό '1' (απενεργοποίηση) για ένα κύκλο πριν απελευθερωθούν.



Σχήμα 2-10. Ο χρονικός συσχετισμός των σημάτων κατά την εγγραφή.

3.1 AMCC S5920 PCI Bridge

3.1.1 ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ S5920

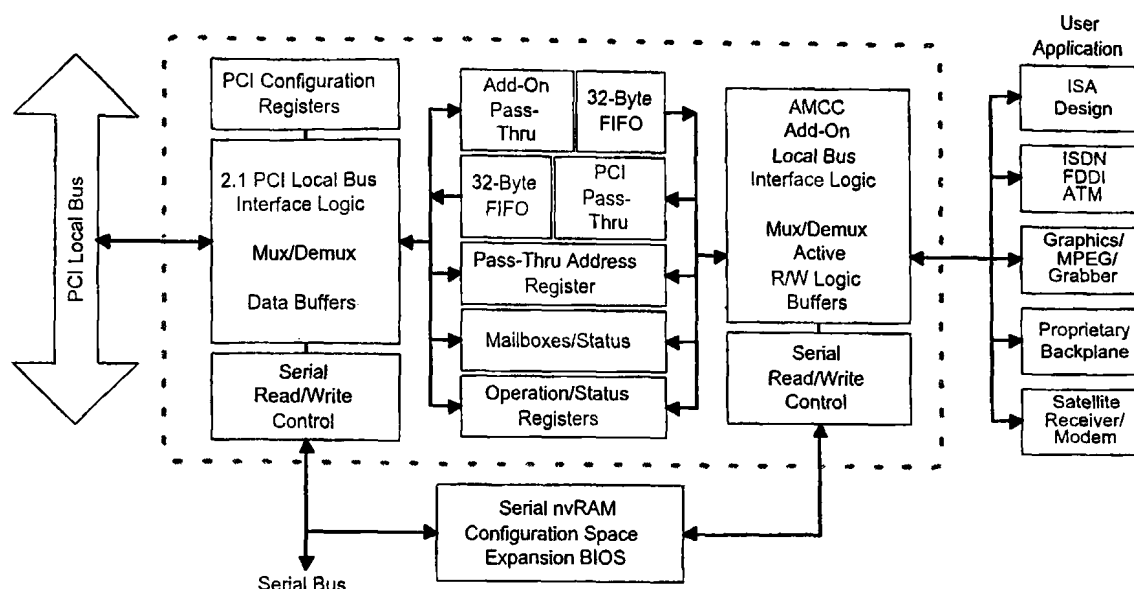
Το ολοκληρωμένο κύκλωμα S5920 αναπτύχθηκε για να παρέχει στον σχεδιαστή μια ολοκληρωμένη και απλή λύση για τον δίαυλο PCI. Χρησιμοποιώντας το S5920 ο σχεδιαστής απαλλάσσεται από τη διαδικασία να καταλάβει τις πολύπλοκες διαδικασίες και την ειδική σηματοδότηση του διαύλου PCI και επίσης δεν χρειάζεται να εξετάσει αν η κατασκευή είναι σύμφωνη με τις προδιαγραφές του διαύλου PCI. [6]

Τα πολύπλοκα σήματα του διαύλου PCI των 33 MHz μετατρέπονται μέσα από το S5920 σε έναν εύχρηστο δίαυλο επικοινωνίας των 8, 16 ή 32 bits, ο οποίος ονομάζεται δίαυλος Add-On (σχήμα 3-1). Ο δίαυλος αυτός επιτρέπει στον χρήστη – σχεδιαστή να υλοποιήσει κυκλώματα που λειτουργούν με μια ανεξάρτητη, από αυτή του PCI, ταχύτητα που μπορεί να φτάσει και τα 40 MHz. Το ολοκληρωμένο S5920 υποστηρίζει μεταφορές δεδομένων μεμονωμένων ή πολλαπλών (burst) κύκλων ρολογιού, που μπορούν να φτάσουν ρυθμούς μετάδοσης μέχρι και 132 Mbyte /s.

Πολλές επιπλέον ιδιότητες του S5920 επιτρέπουν στον χρήστη να υλοποιήσει απλούστερες εφαρμογές τόσο σε Hardware όσο και σε Software. Υπάρχουν μέχρι και τέσσερις ζώνες που μπορούν να καθοριστούν ως παροχές μνήμης ή περιοχές εισόδου / εξόδου (I/O) και αναφέρονται ως περιοχές Pass-Thru.

Δύο καταχωρητές Mailbox των 32 bits έχουν υλοποιηθεί για την μεταφορά επιπλέον δεδομένων ή πληροφοριών κατάστασης και εντολοδότησης. Οι καταχωρητές αυτοί είναι προσπελάσιμοι τόσο από τη μεριά του διαύλου PCI όσο και από τη μεριά του Add-On διαύλου.

Το ολοκληρωμένο S5920 υποστηρίζει τη σύνδεση μιας μνήμης ROM σειριακής επικοινωνίας. Η μνήμη αυτή επιτρέπει στον χρήστη να παραμετροποιήσει τη συμπεριφορά και τη λειτουργία του S5920 κατά την εκκίνηση μετά την παροχή τροφοδοσίας.



Σχήμα 3-1. Το δομικό διάγραμμα του S5920

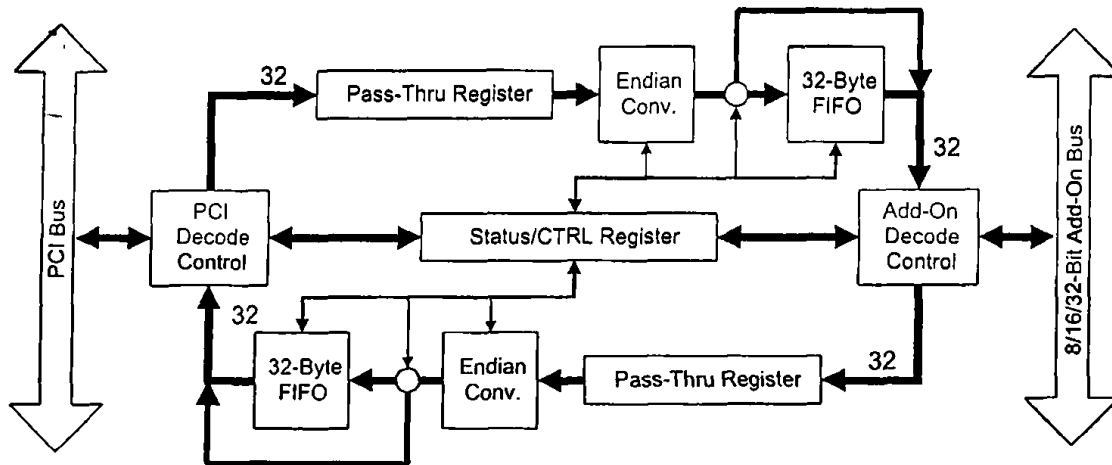
3.2 ΛΕΙΤΟΥΡΓΙΑ PASS-THRU

Κατά τη λειτουργία Pass-Thru εκτελούνται κύκλοι PCI σε πραγματικό χρόνο ή μέσω ενός καταχωρητή FIFO (first in – first out). Η λειτουργία σε πραγματικό χρόνο επιτρέπει στο δίαυλο PCI να αναγνώσει ή να γράψει δεδομένα απευθείας από / στον δίαυλο Add-on.

Το ολοκληρωμένο S5920 επιτρέπει στον σχεδιαστή να υλοποιήσει μέχρι και 4 ανεξάρτητες περιοχές Pass-Thru. Κάθε περιοχή μπορεί να είναι εύρους 8, 16 ή 32 bits και μπορεί να αντιστοιχηθεί σε χώρο μνήμης ή I/O καταλαμβάνοντας όγκο πληροφοριών μέχρι και 512 MB. Στο σχήμα 3-2 εικονίζεται ένα βασικό δομικό διάγραμμα της αρχιτεκτονικής Pass-Thru του S5920.

Οι μεταφορές δεδομένων από τον H/Y (host) προς το κανάλι δεδομένων Pass-Thru του S5920 χρησιμοποιούν συγκεκριμένα pin του Add-On διαύλου για να απεικονίσουν αν έχει ζητηθεί διαδικασία εγγραφής ή ανάγνωσης από τον δίαυλο PCI. Ο χρήστης πρέπει να υλοποιήσει μια εξωτερική λογική (ψηφιακό ηλεκτρονικό κύκλωμα) για να εξακριβώσει (αποκωδικοποιήσει) αν πρέπει να αναγνώσει ή να γράψει δεδομένα στον δίαυλο Add-On του S5920, έτσι ώστε να ικανοποιηθεί το αίτημα που τέθηκε από τον δίαυλο PCI. Τα δεδομένα που θα αποκωδικοποιηθούν περιλαμβάνουν πληροφορίες για το τον τύπο της συναλλαγής (εγγραφή ή ανάγνωση), το εύρος των δεδομένων (8, 16 ή 32 bits), την περιοχή Pass-Thru που ενεργοποιείται, και τέλος τη μέθοδο ανταλλαγής δεδομένων (ενός ή πολλαπλών κύκλων λειτουργίας).

Η λειτουργία Pass-Thru υποστηρίζει κύκλους δεδομένων PCI (PCI data cycles) τόσο μεμονωμένους, όσο και πολλαπλούς (PCI data bursts). Κατά τη λειτουργία πολλαπλών κύκλων δεδομένων (PCI burst operation) το S5920 είναι ικανό να μεταφέρει δεδομένα με το μέγιστο ρυθμό μετάδοσης του PCI. Στην περίπτωση που η εξωτερική λογική που υλοποίησε ο χρήστης είναι αργή, το S5920 θα ζητήσει επανάληψη από το PCI bus μέχρι να ολοκληρωθεί η ζητούμενη μεταφορά δεδομένων.



Σχήμα 3-2 . Το δομικό διάγραμμα της περιοχής Pass-Thru..

Για την βελτιστοποίηση της ανταλλαγής δεδομένων, το κανάλι δεδομένων Pass-Thru περιλαμβάνει και δύο καταχωρητές FIFO των 32 bytes. Ο πρώτος καταχωρητής ασχολείται με τα δεδομένα που διαβάζονται από τον δίαυλο PCI, ενώ ο δεύτερος με τα δεδομένα που γράφονται προς το δίαυλο PCI.

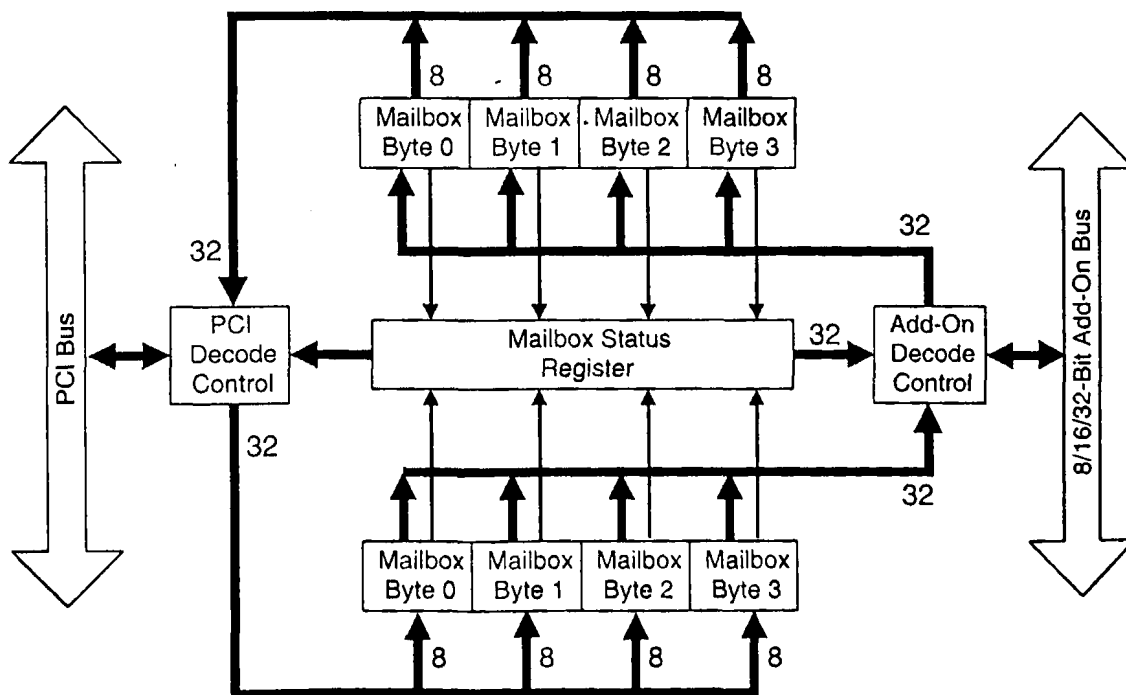
Ο καταχωρητής FIFO ανάγνωσης επιτρέπει στο S5920 να μην παρεμβάλλει καταστάσεις αναμονής (wait states) κατά την ανάγνωση από το PCI αδιαφορώντας για την ταχύτητα λειτουργίας της εξωτερικής λογικής που έχει υλοποιηθεί στον δίαυλο Add-On. Ο καταχωρητής FIFO εγγραφής βοηθά στην προετοιμασία των δεδομένων (Prefetch) από τον δίαυλο Add-On προς τον δίαυλο PCI. Οι καταχωρητές FIFO μπορούν να ενεργοποιηθούν μεμονωμένα για την καλύτερη δυνατή απόδοση.

Ο δίαυλος Add-On μπορεί να λειτουργήσει με 2 διαφορετικούς τρόπους: τον ενεργό και τον παθητικό. Κατά την παθητική λειτουργία το εξωτερικό κύκλωμα του χρήστη οδηγεί τα απαραίτητα σήματα για την ανάγνωση ή την εγγραφή δεδομένων. Κατά την ενεργό λειτουργία ο δίαυλος Add-On οδηγείται από μια εσωτερική μηχανή καταστάσεων του S5920 για τη μείωση των απαιτούμενων εξωτερικών στοιχείων.

3.3 ΛΕΙΤΟΥΡΓΙΑ MAILBOX

Το ολοκληρωμένο S5920 έχει δύο καταχωρητές mailbox των 32 bits. Αυτοί οι καταχωρητές είναι χρήσιμοι για την ανταλλαγή εντολών και πληροφοριών κατάστασης (STATUS) μεταξύ του PCI και του διαύλου Add-On. Από τη μεριά του PCI υπάρχει ένας καταχωρητής (γραμματοκιβώτιο) εισερχομένων και ένας εξερχόμενων. Από την Add-On μεριά υπάρχει ένας καταχωρητής εισερχομένων και ένας εξερχόμενων.

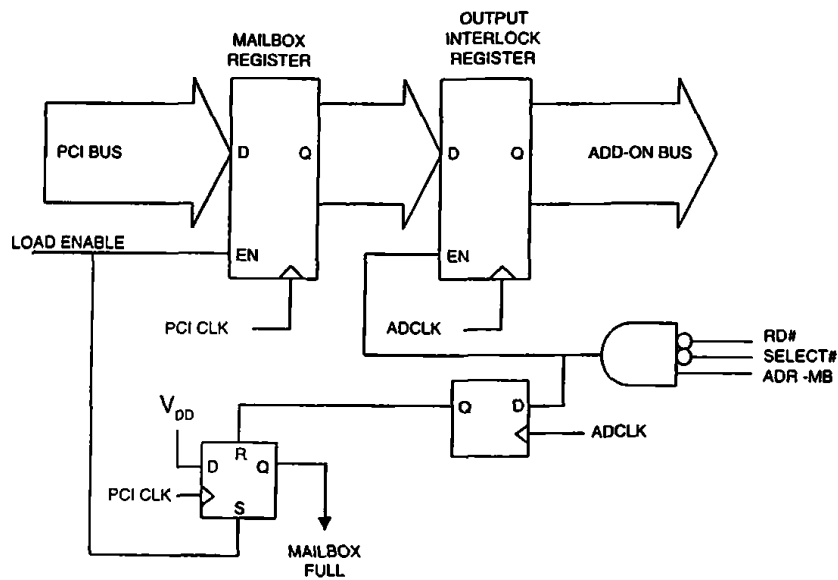
Στην πραγματικότητα ο καταχωρητής εξερχόμενων του PCI ταυτίζεται με τον καταχωρητή εισερχομένων του Add-On και αντίστοιχα ο καταχωρητής εξερχόμενων του ADDOn με τον καταχωρητή εισερχομένων του PCI. Η κατάσταση των «γραμματοκιβωτίων» (mailboxes) μπορεί να εποπτευτεί με 2 τρόπους. Και οι δύο διεπαφές PCI και Add-On έχουν από έναν καταχωρητή κατάστασης γραμματοκιβωτίου (mailbox status register) για τον έλεγχο των δεδομένων στα γραμματοκιβώτια.



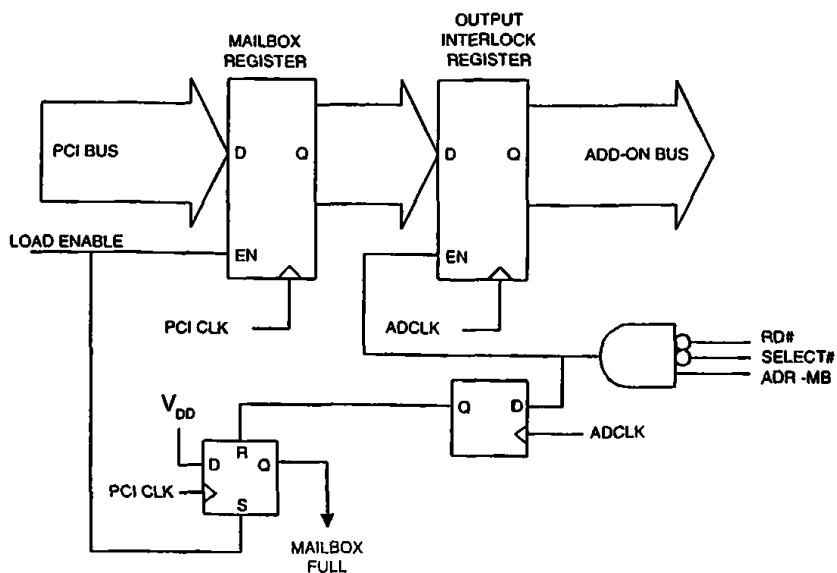
Σχήμα 3-3 . Το δομικό διάγραμμα των καταχωρητών γραμματοκιβωτίου

3.3.1 Περιγραφή Λειτουργίας

Το σχήμα 3-4 δείχνει ένα δομικό διάγραμμα των καταχωρητών γραμματοκιβωτίου PCI και Add-On. Κατά την ανάγνωση του καταχωρητή εισερχομένων του Add-On γραμματοκιβωτίου τα δεδομένα περνούν σε έναν καταχωρητή μανδάλωσης (interlock register). Με τον τρόπο αυτό αποφεύγεται η αλλοίωση των δεδομένων κατά την εγγραφή από το διάυλο PCI στον καταχωρητή εξερχόμενων του PCI.



Σχήμα 3-4 . Εγγραφή mailbox από τη διεπαφή PCI στη διεπαφή Add-On



Σχήμα 3-5 . Εγγραφή mailbox από τη διεπαφή Add-On στη διεπαφή PCI.

Στο σχήμα 3-5 εικονίζεται το δομικό διάγραμμα της εγγραφής από το Add-On στο γραμματοκιβώτιο PCI. Τα δεδομένα που διαβάζονται από το PCI περνούν επίσης από έναν καταχωρητή μανδάλωσης. Με τον τρόπο αυτό αποφεύγεται η καταστροφή των δεδομένων κατά την εγγραφή του γραμματοκιβωτίου, Add-On, εξερχόμενων.

3.3.2 Καταστάσεις ΓΕΜΑΤΟΥ/ ΑΔΕΙΟΥ Γραμματοκιβωτίου

Τόσο η διεπαφή PCI όσο και η Add-On έχουν από έναν καταχωρητή κατάστασης γραμματοκιβωτίου. Οι καταχωρητές MBEF (Mailbox Empty/ Full) και AMBEF δείχνουν την κατάσταση όλων των καταχωρητών του γραμματοκιβωτίου. Το γραμματοκιβώτιο εξερχόμενων της μιας διεπαφής είναι το γραμματοκιβώτιο εισερχομένων για την άλλη διεπαφή, όπως προαναφέραμε. Επομένως, τα bits κατάστασης του εισερχομένου γραμματοκιβωτίου για τον καταχωρητή MBEF είναι πανομοιότυπα με τα bits κατάστασης του εξερχόμενου γραμματοκιβωτίου για τον καταχωρητή AMBEF.

Ο πίνακας 3-1 που ακολουθεί δείχνει τη σχέση των καταχωρητών γραμματοκιβωτίου μεταξύ των διεπαφών PCI και Add-On.

Διεπαφή PCI		Διεπαφή Add-On
Γραμματοκιβώτιο Εξερχομένων	=	Γραμματοκιβώτιο Εισερχομένων
Γραμματοκιβώτιο Εισερχομένων	=	Γραμματοκιβώτιο Εξερχομένων
Καταχωρητής κατάστασης MBEF	=	Καταχωρητής κατάστασης AMBEF

Πίνακας 3-1

Μια διαδικασία εγγραφής στο γραμματοκιβώτιο εξερχόμενων γράφει επίσης δεδομένα στο γραμματοκιβώτιο εισερχομένων της άλλης διεπαφής. Οι σημαίες κατάστασης των καταχωρητών MBEF και AMBEF μηδενίζονται όταν τα αντίστοιχα byte διαβάζονται από το αντίστοιχο γραμματοκιβώτιο εισερχομένων.

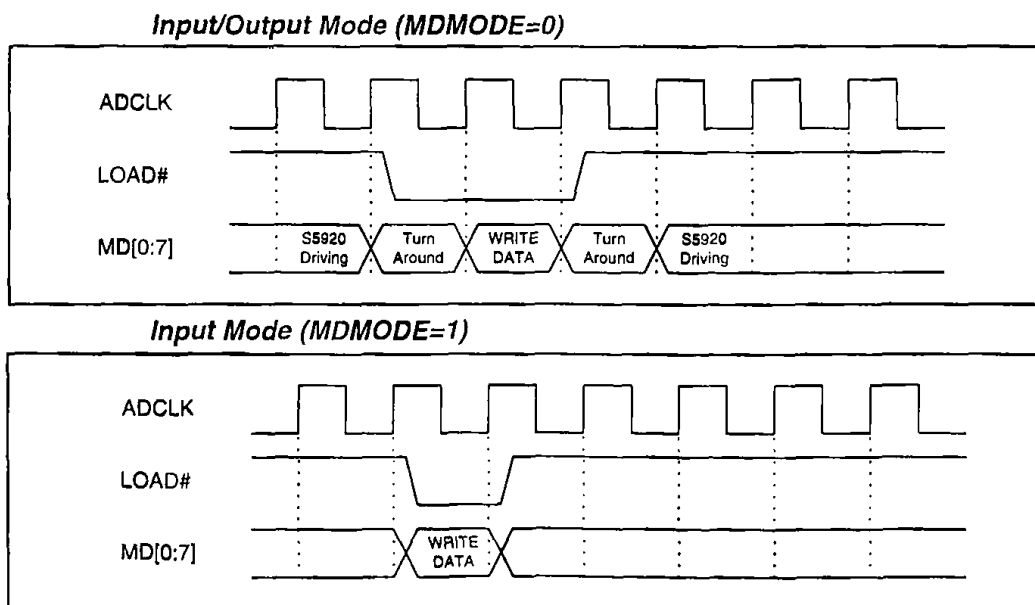


3.3.3 Εναλλακτική πρόσβαση καταχωρητών Mailbox

Το ολοκληρωμένο S5920 είναι εφοδιασμένο με ένα πρόσθετο δίαυλο, ο οποίος επιτρέπει την πρόσβαση του υψηλότερου byte των καταχωρητών mailbox. Η διεπαφή αυτή των 8 bit μπορεί να χρησιμοποιηθεί ξεχωριστά ή και μαζί με τη συνηθισμένη διαδικασία ανάγνωσης των καταχωρητών mailbox από το Add-On bus.

• Ο δίαυλος MD[7:0] (Mailbox Direct) μπορεί να λειτουργήσει με δύο δυνατούς τρόπους: είτε ως έξοδος ή ως αμφίδρομη θύρα επικοινωνίας. Ο τρόπος λειτουργίας καθορίζεται από την κατάσταση του pin MDMODE. Όταν το pin MDMODE είναι σε λογικό "0", τότε ο δίαυλος MD λειτουργεί αμφίδρομα, ενώ όταν το pin MDMODE είναι σε λογικό "1", τότε ο δίαυλος MD[7:0] λειτουργεί ως έξοδος. Στο σχήμα 3-6 εικονίζεται η χρονική σχέση των σημάτων για το Mailbox Direct Bus και για τους δύο τρόπους λειτουργίας. Το pin LOAD# είναι βοηθητικό, και είναι αρνητικής λογικής, ενεργοποιείται δηλαδή με λογικό "0".

Επειδή το υψηλότερο Byte του Mailbox είναι προσπελάσιμο από δύο διαύλους (το δίαυλο MD[7:0] και τον δίαυλο Add-On Da[31:0]), έχει πάντα αποθηκευμένα τα δεδομένα της τελευταίας διαδικασίας εγγραφής ανεξαρτήτως από ποιον από τους δύο διαύλους έγινε.



Σχήμα 3-6. Η χρονική συσχέτιση των σημάτων για τους 2 τρόπους λειτουργίας της διεπαφής Mailbox Direct

3.4 Καταχωρητές λειτουργίας του S5920

Όλες οι λειτουργίες έλεγχου και επικοινωνίας του S5920, εκτελούνται μέσω 2 ομάδων καταχωρητών. Τους καταχωρητές λειτουργίας για το δίαυλο PCI και τους καταχωρητές λειτουργίας για το δίαυλο Add-On. Κάποιοι καταχωρητές και των 2 ομάδων είναι προσπελάσιμοι και από τους 2 διαύλους. Ταυτόχρονη προσπέλαση των κοινών αυτών καταχωρητών και από τους 2 διαύλους δεν ενδείκνυται και μπορεί να οδηγήσει σε απρόβλεπτη συμπεριφορά του ολοκληρωμένου S5920.

Στους πίνακες 3-2 και 3-3, εμφανίζονται οι καταχωρητές λειτουργίας και για τους 2 διαύλους. Στη συνέχεια αναλύονται μόνο οι καταχωρητές που έχουν σχέση με τη λειτουργία mailbox. Η περαιτέρω εμβάθυνση ξεφεύγει από τα όρια της εργασίας αυτής, για αυτό θα περιοριστούμε στους καταχωρητές, που έχουν σχέση με την συγκεκριμένη υλοποίηση της κάρτας PCI.

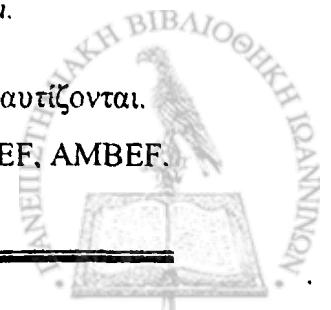
Address Offset	Συντομογραφία	Όνομασία Καταχωρητή
0Ch	OMB	Outgoing Mailbox Register
1Ch	IMB	Incoming Mailbox Register
34h	MBEF	Mailbox Empty/Full Status Register
38h	INTSCR	Interrupt Control/Status Register
3Ch	RCR	Reset Control Register
60h	PTCR	Pass-Thru Configuration Register.

Πίνακας 3-2 Οι καταχωρητές λειτουργίας για τον δίαυλο PCI.

Address Offset	Συντομογραφία	Όνομασία Καταχωρητή
0Ch	AIMB	Add-On Incoming Mailbox Register
1Ch	AOMB	Add-On Outgoing Mailbox Register
28h	APTA	Add-On Pass-Thru Address Register.
2Ch	APTD	Add-On Pass-Thru Data Register.
34h	AMBEF	Add-On Mailbox Empty/Full Status Register
38h	AINT	Add-On Interrupt Control/Status Register
3Ch	ARCR	Add-On Reset Control Register
60h	APTCR	Add-On Pass-Thru Configuration Register.

Πίνακας 3-3. Οι καταχωρητές λειτουργίας για τον δίαυλο Add-On.

Όπως φαίνεται και από τους πίνακες οι καταχωρητές OMB και AIMB ταυτίζονται. Το ίδιο ισχύει και για τους καταχωρητές IMB, AOMB και για τους MBEF, AMBEF.



□ Καταχωρητής OMB

Ο καταχωρητής αυτός των 32 bits χρησιμεύει για την αποστολή εντολών και παραμέτρων λειτουργίας στο δίαυλο Add-On. Οι μεταφορές δεδομένων μέσω του καταχωρητή OMB μπορούν να γίνουν σε επίπεδο 8,16 ή 32 bits. Ο καταχωρητής αυτός ονομάζεται επίσης AOMB.

□ Καταχωρητής IMB

Ο καταχωρητής αυτός των 32 bits χρησιμεύει για την λήψη εντολών και παραμέτρων λειτουργίας από το δίαυλο Add-On. Οι μεταφορές δεδομένων μέσω του καταχωρητή IMB μπορούν να γίνουν σε επίπεδο 8,16 ή 32 bits. Ο καταχωρητής αυτός ονομάζεται επίσης AOMB.

□ Καταχωρητής MBEF

Ο καταχωρητής αυτός παρέχει εποπτεία για την κατάσταση κάθε καταχωρητή mailbox. Η κατάσταση του καταχωρητή OMB (άδειος / γεμάτος) απεικονίζεται στα bits 15 έως 12 ενώ για τον καταχωρητή IMB στα bits 31 έως 28 όπως προκύπτει από τον πίνακα 3-4. Συγκεκριμένα λογικό '1' σε κάποιο από τα bits δηλώνει ότι το αντίστοιχο byte είναι πλήρες. Τα υπόλοιπα bit του καταχωρητή είναι δεσμευμένα από τον κατασκευαστή.

MBEF	Σχετιζόμενο byte
Bit 31	IMB byte 3
Bit 30	IMB byte 2
Bit 29	IMB byte 1
Bit 28	IMB byte 0
Bit 15	OMB byte 3
Bit 14	OMB byte 2
Bit 13	OMB byte 1
Bit 12	OMB byte 0

Πίνακας 3-4. Η αντιστοιχία των status bits του MBEF με την πληρότητα των mailbox.

4. Γέφυρες I²C

Όπως αναφέρθηκε και στην παράγραφο 1.2, συσκευές master για το δίαυλο I²C μπορούν να είναι μικροεπεξεργαστές ή μικροελεγκτές. Οι συσκευές αυτές μπορεί να έχουν ειδικές εσωτερικές διατάξεις για να παράγουν σήματα I²C (διεπαφές I²C) ή μπορεί να εξομοιώνουν τη λειτουργία του διαύλου I²C με software χρησιμοποιώντας δύο από τα πολλά pins, εισόδου/εξόδου που έχουν, για τις γραμμές SDA και SCL.

- Είναι προφανές πως οι μικροϋπολογιστές που εξομοιώνουν τη λειτουργία του διαύλου I²C μέσω software δεν μπορούν να παράγουν σήματα ρολογιού πολύ μεγάλης συχνότητας. Επιπλέον, τα σήματα αυτά δεν έχουν σταθερή περίοδο. Το πρώτο οφείλεται στο γεγονός ότι ο κώδικας του software δεν εκτελείται ικανοποιητικά γρήγορα, ενώ το δεύτερο στο γεγονός ότι οι εντολές των μικροϋπολογιστών δεν εκτελούνται όλες στον ίδιο χρόνο.

Από όσα παρατέθηκαν παραπάνω συσκευές master είναι συνήθως κάποιες «έξυπνες» συσκευές που έχουν έτοιμες διεπαφές ή εξομοιώνουν τη λειτουργία του διαύλου. Οι συσκευές αυτές μπορούν να χαρακτηριστούν ως γέφυρες I²C κατά όμοιο τρόπο όπως το ολοκληρωμένο S5920 χαρακτηρίζεται ως γέφυρα PCI.

Η εταιρεία PHILIPS έχει να προσθέσει ένα ακόμη ολοκληρωμένο κύκλωμα στην κατηγορία των γεφυρών I²C. Το ολοκληρωμένο αυτό ονομάζεται PCF8584 και αποτελεί έναν ελεγκτή διαύλου I²C. [7]

Το PCF8584 δεν είναι ένας μικροϋπολογιστής, αλλά περιέχει την ειδική διεπαφή I²C που θα μπορούσε να έχει ένας μικροελεγκτής. Στις παραγράφους που ακολουθούν παρατίθεται αναλυτικότερα η λειτουργία του.

4.1. Γενική περιγραφή του PCF8584

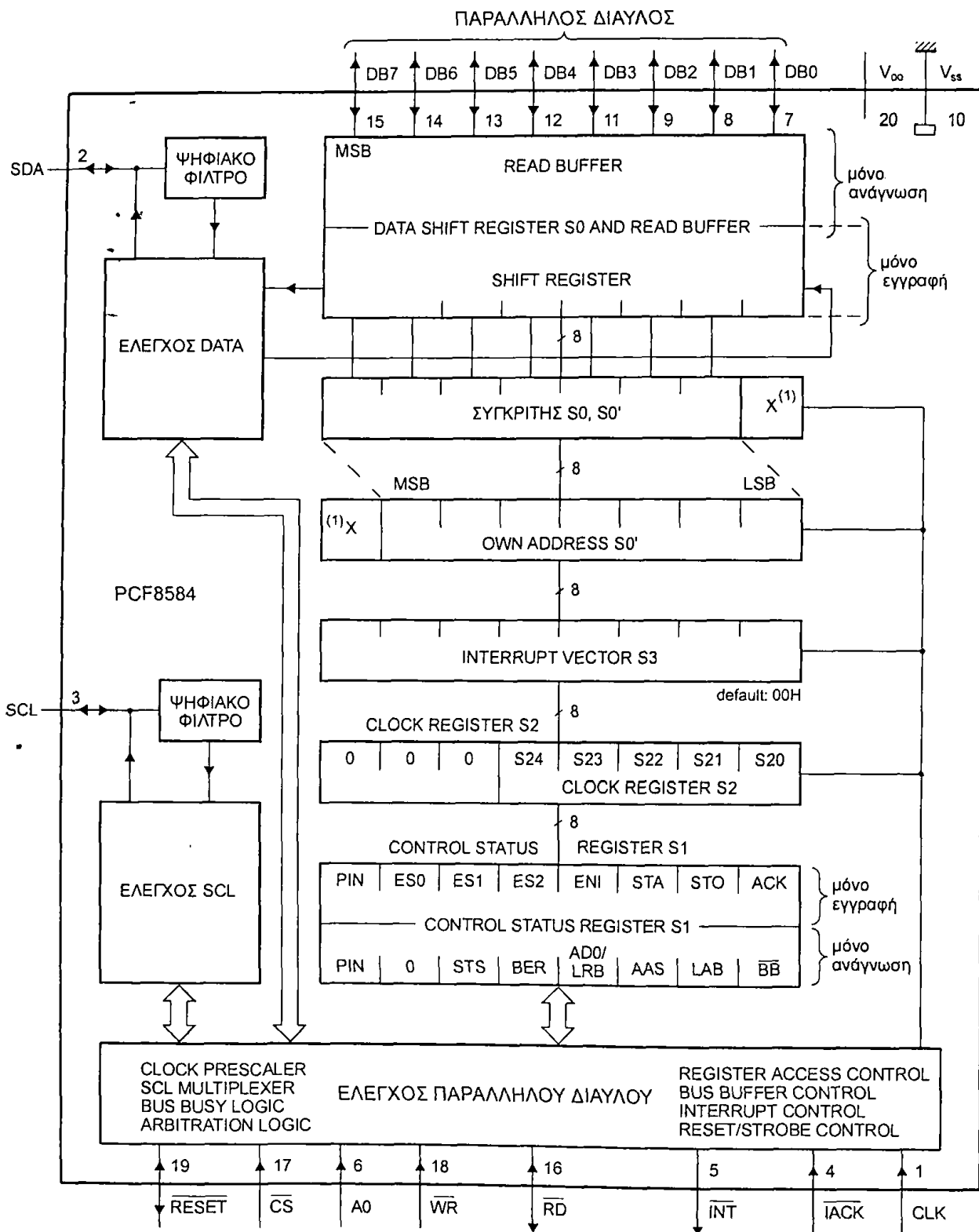
Το ολοκληρωμένο κύκλωμα PCF8584 τεχνολογίας CMOS λειτουργεί ως διεπαφή μεταξύ μικροϋπολογιστών με παράλληλες θύρες επικοινωνίας και του διαύλου I²C. Το PCF8584 υποστηρίζει τόσο λειτουργία master όσο και slave και είναι σύμφωνο με τις προδιαγραφές του I²C (standard mode). Το PCF8584 επιτρέπει σε συστήματα παράλληλων διαύλων να επικοινωνήσουν αμφίδρομα με το δίαυλο I²C.

Ένα τέτοιο σύστημα αποτελεί και η κάρτα PCI που αναλύεται στην παρούσα εργασία.

Όπως φαίνεται και από το δομικό διάγραμμα (σχήμα 4-1), το ολοκληρωμένο PCF8584 περιέχει εσωτερικά:

- 1) έναν καταχωρητή ολίσθησης τον S0,
- 2) έναν καταχωρητή S0', όπου ο χρήστης μπορεί να αποθηκεύσει την τιμή της διεύθυνσης slave του PCF8584
- 3) έναν συγκριτή, που συγκρίνει τα περιεχόμενα των S0 και S0', 4) έναν καταχωρητή για το άνυσμα διακοπής S3, 5) έναν καταχωρητή ρυθμίσεων ρολογιού S2, και τέλος έναν καταχωρητή control / status S1.

Οι καταχωρητές S0, S1, S2, S3 και S4 γράφονται και διαβάζονται μέσω του Data Bus DB0 έως DB7, ενώ η λειτουργία του PCF8584 καθορίζεται από τους ακροδέκτες ελέγχου του παράλληλου διαύλου.



(1)χ=Αδιάφορο

Σχήμα 4-1. Το δομικό διάγραμμα του PCF8584

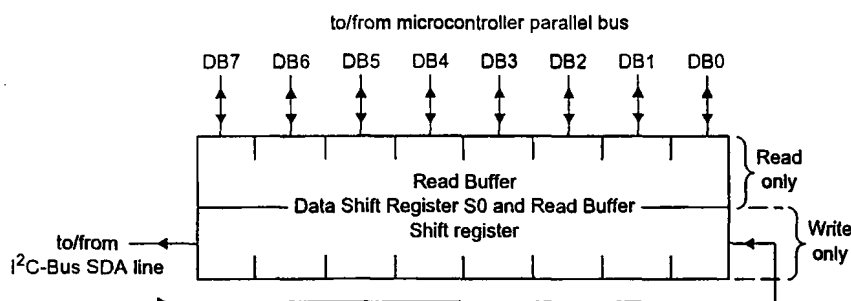


4.2 Καταχωρητές

Στη συνέχεια ακολουθεί ανάλυση των εσωτερικών καταχωρητών του ολοκληρωμένου κυκλώματος PCF8584.

4.2.1 Καταχωρητής ολίσθησης S0

Ο καταχωρητής S0 λειτουργεί ως ένας σειριακός καταχωρητής ολίσθησης, ο οποίος μπορεί να φορτωθεί από τον παράλληλο δίαυλο [DB0:DB7]. Όλες οι διαδικασίες ανάγνωσης και εγγραφής στο δίαυλο I²C γίνονται από τον καταχωρητή S0.



Σχήμα 4-2. Ο καταχωρητής Ολίσθησης S0

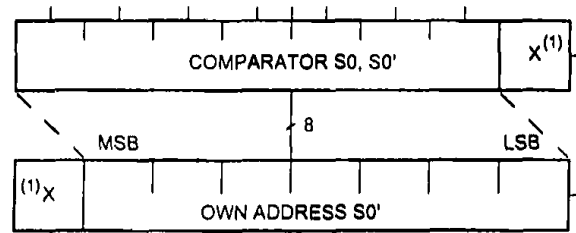
4.2.2 Καταχωρητής S0' (Own slave address)

Όταν το PCF8584 λειτουργεί ως συσκευή slave, ο καταχωρητής αυτός πρέπει να περιέχει την τιμή της διεύθυνσης slave στην οποία ο PCF8584 θα απαντήσει. Η τιμή του καταχωρητή αποθηκεύεται κατά την αρχικοποίηση του PCF8584, ασχέτως αν θα χρησιμοποιηθεί ή όχι.

4.2.3 Συγκριτής

Κατά τη λειτουργία slave, η διεύθυνση που εκπέμπεται από τον master του διαύλου συγκρίνεται με την τιμή του καταχωρητή S0'. Αν οι τιμές τους συμπίπτουν, τότε η σημαία (AAS) του καταχωρητή S1 πηγαίνει σε λογικό '1'.

Όπως φαίνεται στο σχήμα 4-3 οι καταχωρητές S0 και S0' είναι μετατοπισμένοι κατά ένα bit. Αυτό σημαίνει ότι αν ο καταχωρητής S0' προγραμματιστεί με την τιμή SSh, η διεύθυνση στην οποία απαντά ως συσκευή slave θα είναι η AAh.



Σχήμα 4-3. Η διάταξη του καταχωρητή S0' και του συγκριτή

4.2.4 Καταχωρητής ρολογιού S2

Από τον καταχωρητή S2 καθορίζεται η συχνότητα λειτουργίας του διαύλου I²C, καθώς επίσης και τιμή διαίρεσης της συχνότητας του σήματος clock που εφαρμόζεται εξωτερικά του PCF8584.

Στους πίνακες που ακολουθούν εικονίζονται οι διάφορες τιμές των bit του S2 για την κατάλληλη επιλογή συχνοτήτων.

BIT		Συχνότητα SCL
S21	S20	[KHz]
0	0	90
0	1	45
1	0	11
1	1	1.5

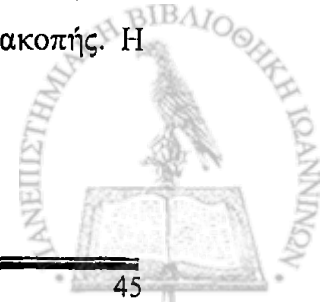
Πίνακας 4-1

BIT			Εξωτερικό Ρολόι
S24	S23	S22	[MHz]
0	X	X	3
1	0	0	4.43
1	0	1	6
1	1	0	8
1	1	1	12

Πίνακας 4-2

4.2.5 Άνυσμα Διακοπής S3

Ο καταχωρητής αυτός περιέχει την προγραμματιζόμενη τιμή του ανύσματος διακοπής για μικροελεγκτές που υποστηρίζουν αυτή τη λειτουργία. Η τιμή του ανύσματος στο δίαυλο [DB0:DB7] κατά τη δημιουργία μιας αίτησης διακοπής. Η προκαθορισμένη τιμή του καταχωρητή αυτού είναι «00h».



4.2.6 Ο Καταχωρητής Control/ Status S1

Ο καταχωρητής S1 ελέγχει όλες τις λειτουργίες του διαύλου I²C και ταυτόχρονα παρέχει πληροφορίες κατάστασης (status).

Υποκαταχωρητής		BITS							Λειτουργία
Control	PIN	ESO	ES1	ES2	ENI	STA	STO	ACK	εγγραφή
Status	PIN	0	STS	BER	LRB	AAS	LAB	BB	ανάγνωση

Πίνακας 4-3

Ο καταχωρητής είναι προσβάσιμος όταν το pin επιλογής A0 (σχήμα 4-1) είναι σε υψηλή λογική κατάσταση. Στην πραγματικότητα αποτελείται από δύο καταχωρητές: ο ένας καταχωρητής είναι μόνο ανάγνωσης (STATUS) και άλλος μόνο εγγραφής (control). Στον πίνακα 4-3 φαίνονται τα επιμέρους bit των καταχωρητών αυτών.

4.2.6.1. Τα bits του καταχωρητή Control S1

Το bit ESO ενεργοποιεί και απενεργοποιεί τη σειριακή διεπαφή I²C, ενώ τα bit ES1 και ES2 επιτρέπουν την πρόσβαση στους υπόλοιπους καταχωρητές, όπως φαίνεται στον πίνακα 4-4.

ΔΙΕΥΘΥΝΣΙΟΔΟΤΗΣΗ ΤΩΝ ΕΣΩΤΕΡΙΚΩΝ ΚΑΤΑΧΩΡΗΤΩΝ				
A0	ES1	ES2	IACK	Λειτουργία
ESO = 0 : σειριακή διεπαφή απενεργοποιημένη				
1	0	X	1	Ανάγνωση/Εγγραφή S1
0	0	0	1	Ανάγνωση/Εγγραφή S0'
0	0	1	1	Ανάγνωση/Εγγραφή S3
0	1	0	1	Ανάγνωση/Εγγραφή S2
ESO = 1 : σειριακή διεπαφή ενεργοποιημένη				
1	0	X	1	Εγγραφή S1 (Control)
1	0	X	1	Ανάγνωση S1 (Status)
0	0	0	1	Ανάγνωση/Εγγραφή S0
0	0	1	1	Ανάγνωση/Εγγραφή S3
X	0	X	0	Ανάγνωση S3

Πίνακας 4-4

Το bit ENI ενεργοποιεί την έξοδο \overline{INT} , το οποίο οδηγείται από την εσωτερική σημαία PIN (λογικό '0').

Τα bits STA και STO ελέγχουν την παραγωγή των διαδικασιών START και STOP, την εκπομπή της διεύθυνσης slave, όπως φαίνεται στον πίνακα 4-5.



STA	STO	Παρούσα Κατάσταση	Λειτουργία	Ενέργεια
1	0	Slave-Αποδέκτης	Start	Αποστολή START + διέθυση παραμονή ως Master-Πομπός αν R/W=0 αλλαγή σε Master-Αποδέκτη αν R/W=1
1	0	Master-Πομπός	Επαναληπτικό Start	Αποστολή START + διέθυση παραμονή ως Master-Πομπός αν R/W=0 αλλαγή σε Master-Αποδέκτη αν R/W=1
0	1	Master Πομπός/Δέκτης	Stop Εκπομπή/Λήψη	Αποστολή STOP αλλαγή σε Slave-Αποδέκτη
1	1	Master	Αλυσιδωτή μεταφορά	Αποστολή STOP μετά START, διεύθυνσης μετά το τελευταίο πλαίσιο του master χωρίς να έχει σταλεί το STOP (συνεχής μεταφορά χωρίς απελευθέρωση του διαύλου)
0	0	Οποιαδήποτε	NOP	Καμία λειτουργία

Πίνακας 4-5

Το bit ACK πρέπει να βρίσκεται σε κανονική λειτουργία σε λογικό '1' και τότε ο ελεγκτής του διαύλου I²C στέλνει αυτόματα επιβεβαίωση μετά από κάθε byte. Όταν γίνει '0', τότε σηματοδοτεί ότι ο master-αποδέκτης δεν περιμένει άλλα δεδομένα από τον slave-πομπό.

4.2.6.2. Οι σημαίες του καταχωρητή κατάστασης S1

Η σημαία PIN χρησιμοποιείται για αιτήσεις διακοπής και οδηγεί την έξοδο \overline{INT} , όταν αυτή έχει ενεργοποιηθεί.

Η σημαία STS οδηγείται σε υψηλή λογική κατάσταση, όταν ανιχνεύεται μια διαδικασία STOP και το PCF8584 λειτουργεί ως slave-αποδέκτης.

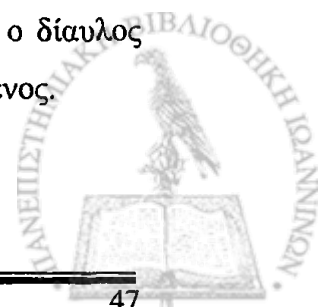
Η σημαία BER (BUS ERROR) ενεργοποιείται όταν δεν ανιχνεύεται σωστά μια διαδικασία START ή STOP.

Η σημαία LRB κρατάει το τελευταίο bit που εκπέμφθηκε στο δίαυλο (συνήθως το ACK που προέρχεται από έναν slave).

Η σημαία AAS ενεργοποιείται όταν ο PCF8584 λειτουργεί σαν slave και πρέπει να απαντήσει στην αίτηση ενός master.

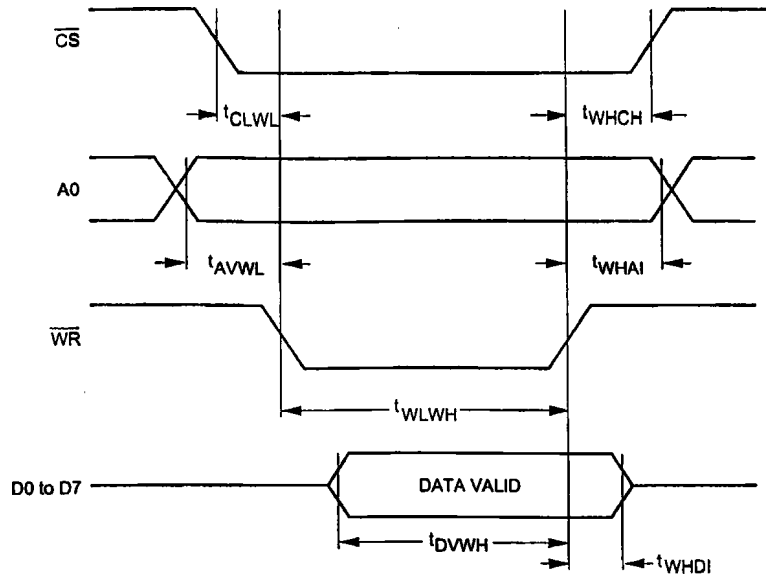
Η σημαία LAB (Lost Arbitration Bit) γίνεται '1' όταν σε ένα δίαυλο που έχει πολλούς master η κυριότητα του διαύλου έχει δεσμευθεί από κάποιον άλλο master.

Η σημαία \overline{BB} (Bus Busy). Η σημαία αυτή δείχνει πότε χρησιμοποιείται ο δίαυλος I²C. Μια χαμηλή λογική κατάσταση δείχνει ότι ο δίαυλος είναι απασχολημένος.

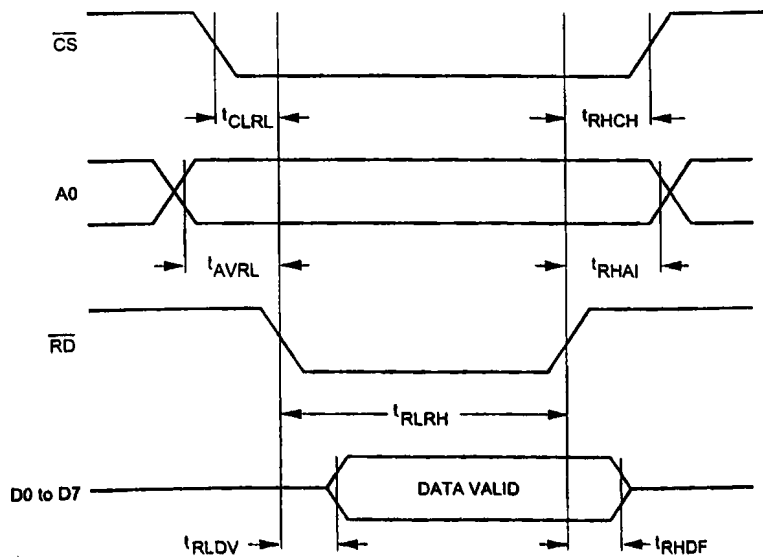


4.3 Χρονικά διαγράμματα

Τα χρονικά διαγράμματα των σημάτων κατά την διάρκεια ενός κύκλου εγγραφής και ενός κύκλου ανάγνωσης φαίνονται στα σχήματα 4 και 5 αντίστοιχα.



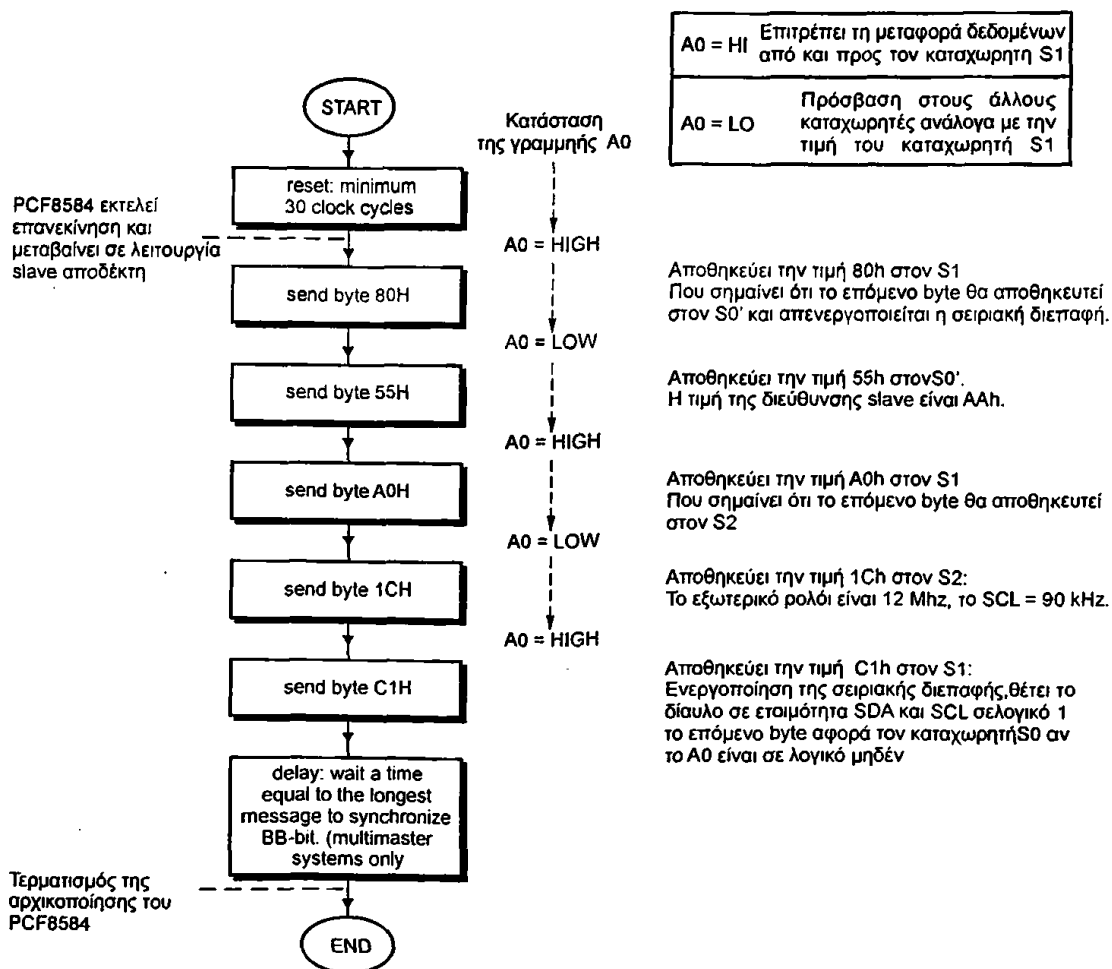
Σχήμα 4-4. Κύκλος εγγραφής του PCF8584.



Σχήμα 4-5. Κύκλος ανάγνωσης του PCF8584.

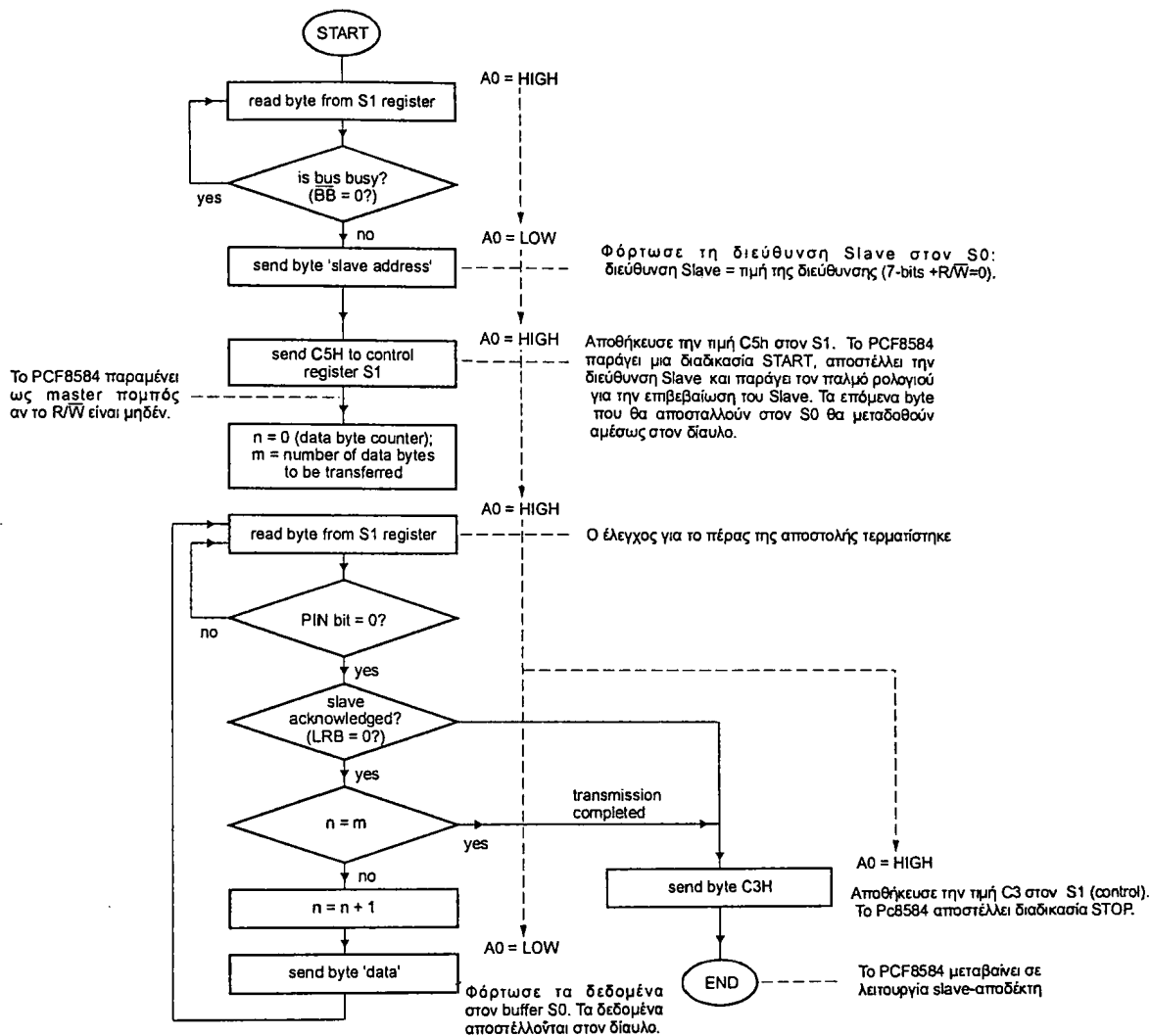
4.4 Software

Το λογισμικό οδήγησης του PCF8584 εξαρτάται μόνο από τον μικροϋπολογιστή που οδηγεί το PCF8584 και μπορεί να γραφεί σε διάφορες γλώσσες προγραμματισμού. Ωστόσο μπορούν να παρατεθούν κάποιες γενικές οδηγίες. Στα σχήματα που ακολουθούν εικονίζονται παραδείγματα διαγραμμάτων ροής για τη δημιουργία κώδικα οδήγησης του PCF8584.



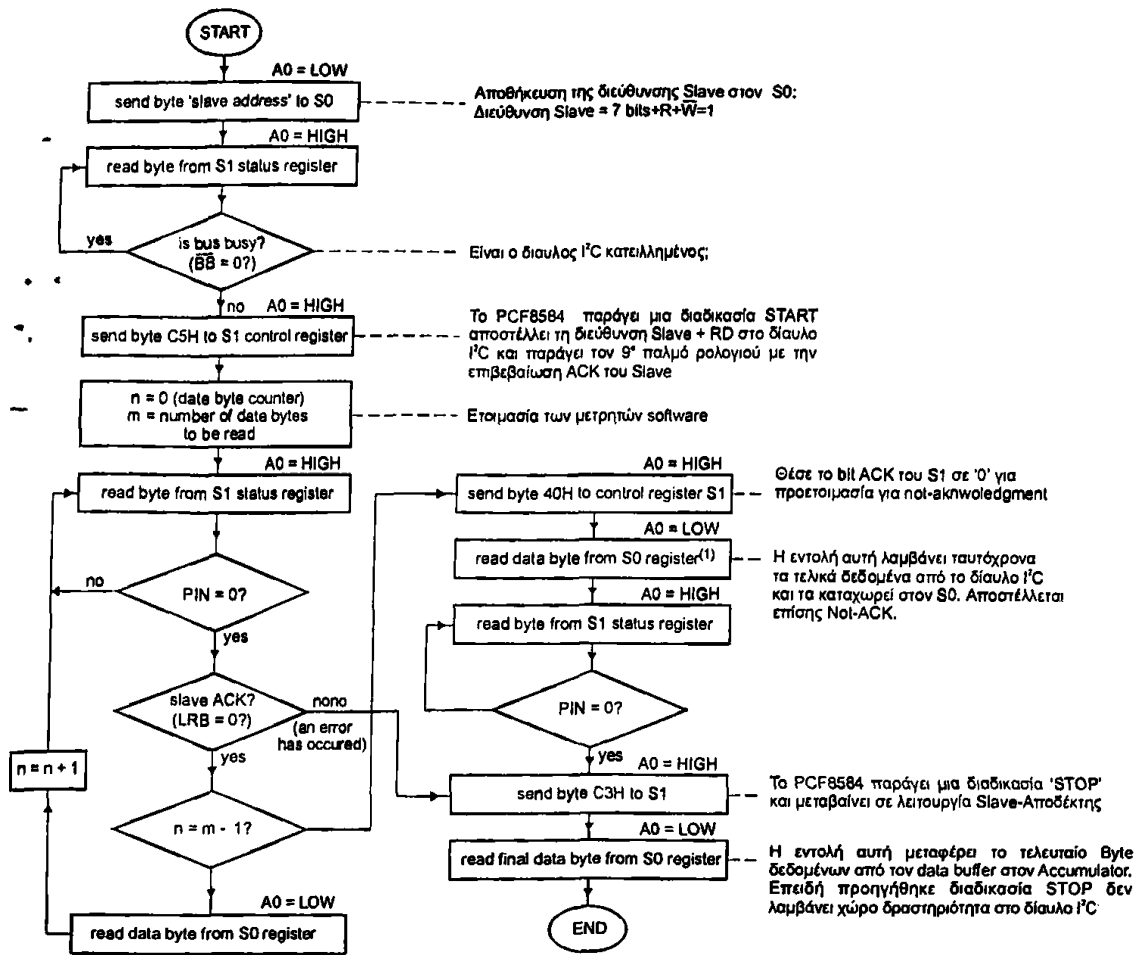
Σχήμα 4-6. Διάγραμμα ροής προγραμματισμού για την αρχικοποίηση του PCF8584.





Σχήμα 4-7. Διάγραμμα ροής λειτουργίας master-πομπού του PCF8584.



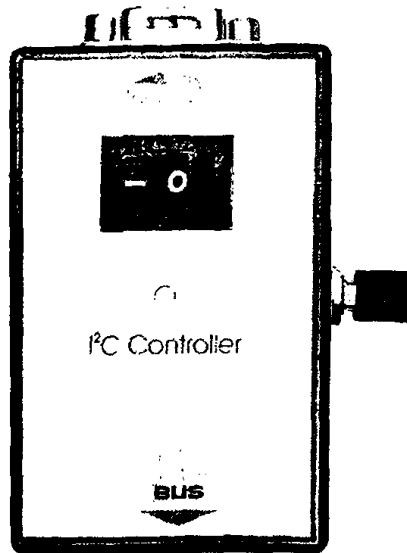


Σχήμα 4-8. Διάγραμμα ροής λειτουργίας master-αποδέκτη του PCF8584.

5. RS-232 I²C Controller

5.1 Σκοπός Υλοποίησης

Στα πλαίσια της διπλωματικής εργασίας αναπτύχθηκε και μία αυτόνομη μονάδα παραγωγής σημάτων I²C, η οποία συνδέεται με ένα Η/Υ μέσω της σειριακής θύρας RS-232. Ο λόγος που υλοποιήθηκε η μονάδα αυτή, ήταν η δοκιμή του ολοκληρωμένου PCF8584, πριν την τελική χρησιμοποίησή του στην κάρτα PCI. Η δοκιμή του ολοκληρωμένου PCF8584, είχε ως σκοπό την διερεύνηση της συμπεριφοράς του κατά τη λειτουργία του, καθώς επίσης και την εξοικείωση του χρήστη με το δίαυλο I²C.



Εικόνα 5-1. Φωτογραφία της πρόσοψής του RS-232 I²C Controller.

Η υλοποίηση της αυτόνομης μονάδας χωρίστηκε σε δύο βήματα: στην ανάπτυξη του hardware και στη δημιουργία software στον Η/Υ. Στις ενότητες που ακολουθούν, θα γίνει εκτενέστερη ανάλυση των προαναφερθέντων.

5.2 Hardware

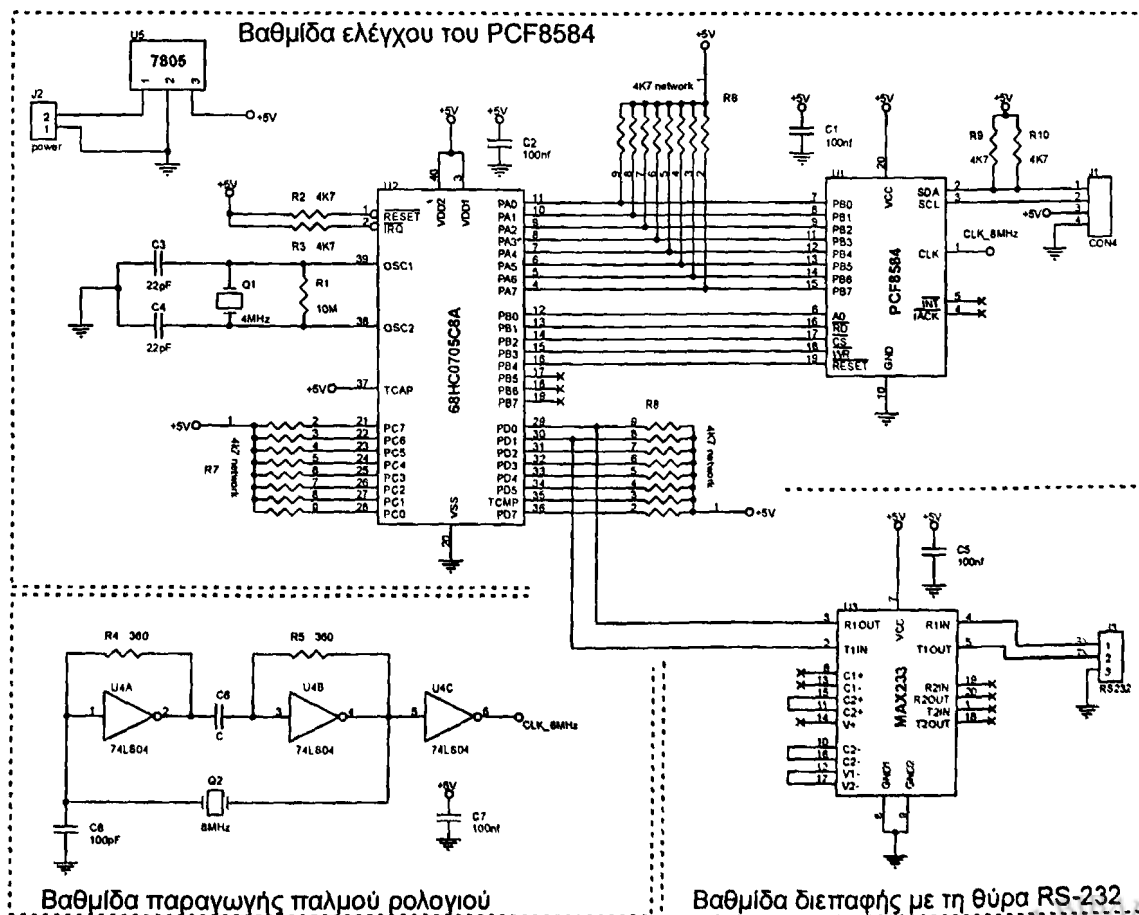
Για τον έλεγχο του PCF8584 χρησιμοποιήθηκε ο μικροελεγκτής MC68HC705C8A της εταιρείας Motorola. Πρόκειται για ένα μικροελεγκτή τεχνολογίας CISC, ο οποίος διαθέτει μνήμη EPROM και μπορεί να προγραμματιστεί πολλές φορές. Ο μικροελεγκτής, διαθέτει τρεις γενικές θύρες εισόδου/ εξόδου των 8 bit, καθώς επίσης μία θύρα σύγχρονης σειριακής επικοινωνίας και μια θύρα ασύγχρονης σειριακής επικοινωνίας. [8]

Το κύκλωμα του RS232-I²C controller μπορεί να χωρισθεί σε 3 τμήματα. Τη βαθμίδα ελέγχου του PCF8584, τη βαθμίδα παραγωγής παλμού ρολογιού για το PCF 8584 και τη βαθμίδα διεπαφής με την σειριακή θύρα RS232. Οι τρεις βαθμίδες εικονίζονται στο ηλεκτρονικό διάγραμμα του σχήματος 1.

5.2.1 Βαθμίδα Ελέγχου

Η βαθμίδα ελέγχου, όπως προαναφέρθηκε, βασίζεται στον μικροελεγκτή MC68HC705C8A. Για τον έλεγχο του PCF8584 χρειάζονται συνολικά 13 ακροδέκτες. Οι οκτώ από αυτούς (PB0 έως PB7) οδηγούνται από την θύρα A του μC, ενώ οι υπόλοιποι 5 (A0, RD, WR, CS, RESET) οδηγούνται από τους ακροδέκτες 0 έως 4 της θύρας B του μC. Η θύρα A, έχει προγραμματιστεί σε αμφίδρομη λειτουργία ενώ η θύρα B προγραμματίστηκε για να λειτουργεί μόνο ως έξοδος. [9]

Το δικτύωμα Q1, C3, C4 και R1 αποτελούν το κύκλωμα παραγωγής του παλμού ρολογιού για τη λειτουργία του μC. Η έξοδος του σήματος I²C, παρέχεται από τον σύνδεσμο J1, ενώ τάση τροφοδοσίας του όλου κυκλώματος παρέχεται στον σύνδεσμο J2 και είναι 7-20 Volts DC.



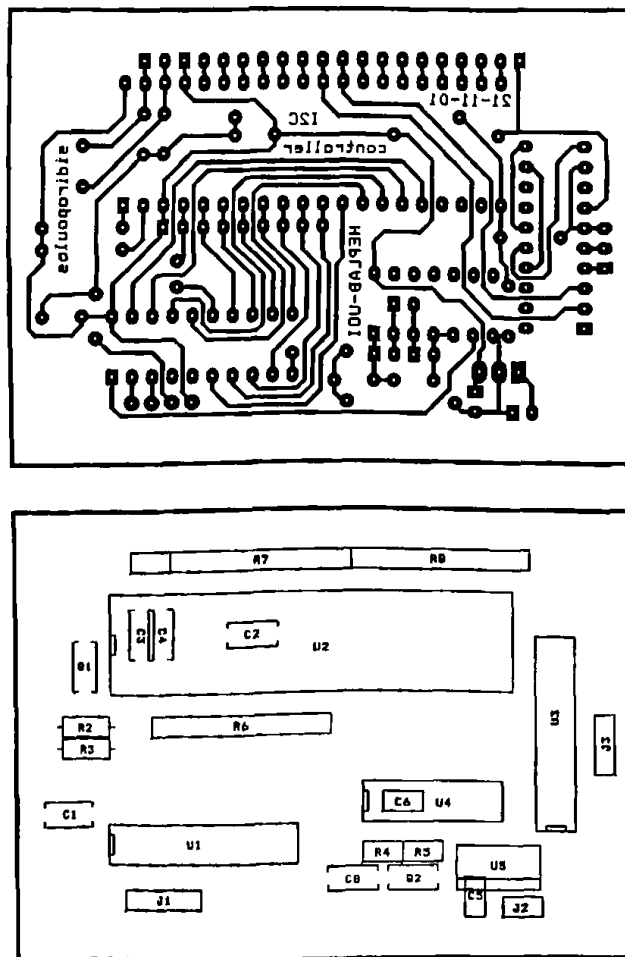
Σχήμα 5-1. Το ηλεκτρονικό κύκλωμα του RS-232 I²C controller.

5.2.2 Βαθμίδα διεπαφής με τη θύρα RS-232

Η βαθμίδα αυτή βασίζεται εξ' ολοκλήρου στο ολοκληρωμένο κύκλωμα MAX233 της εταιρίας MAXIM. Το ολοκληρωμένο αυτό λειτουργεί ως μεταφραστής επιπέδων μεταξύ των επιπέδων (TTL) της σειριακής θύρας του μC και των επιπέδων τάσης σύμφωνα με τα οποία λειτουργεί η θύρα RS-232. Η σύνδεση με τον ηλεκτρονικό υπολογιστή γίνεται μέσω του συνδέσμου J3.

5.2.3 Βαθμίδα παραγωγής παλμού ρολογιού

Για τη λειτουργία του PCF8584, χρειάζεται παλμός ρολογιού συχνότητας 8 MHz, ο οποίος παράγεται από το δικτύωμα των πυλών U4A, U4B, U4C του κρυστάλλου Q2 και του πυκνωτή C6. [10]



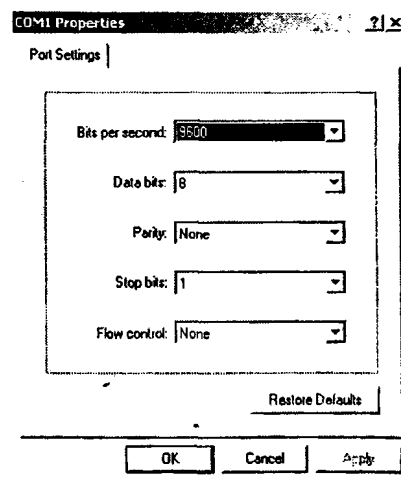
Σχήμα 5-2. Το τυπωμένο κύκλωμα του RS-232 I²C controller και το τοπογραφικό τοποθέτησης των ηλεκτρονικών εξαρτημάτων.

5.3 Software

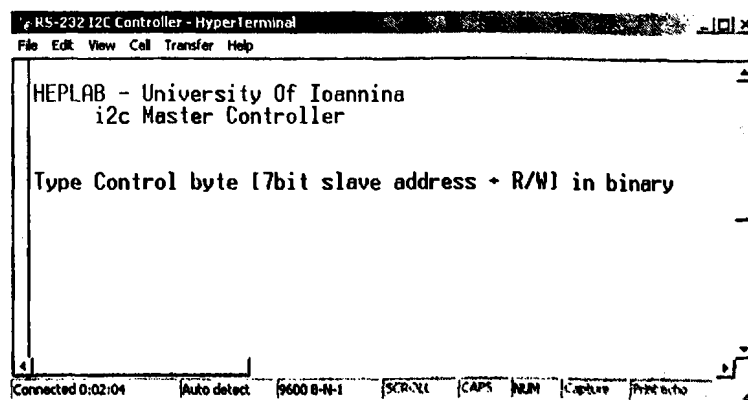
Ο RS-232 I²C controller, υποστηρίζει δύο interfaces χρήσης για τον Η/Υ. Το ένα από αυτά μπορεί να είναι οποιοδήποτε πρόγραμμα τύπου terminal για το χειρισμό της σειριακής θύρας (π.χ. Hyper Terminal). Το άλλο πρόγραμμα έχει γραφεί σε γλώσσα G χρησιμοποιώντας το πακέτο LabVIEW.

5.3.1 Έλεγχος του I²C Controller μέσω Hyper Terminal

Για να συνεργαστεί η συσκευή με το Hyper Terminal ο χρήστης δεν έχει παρά να ξεκινήσει το πρόγραμμα και να επιλέξει απευθείας επικοινωνία με την θύρα Com στην οποία έχει συνδεθεί εξωτερικά ο RS-232 I²C Controller. Στη συνέχεια, στα χαρακτηριστικά της σύνδεσης πρέπει να επιλεγεί ρυθμός μεταφοράς δεδομένων 96000 bps, μήκος δεδομένων 8-bit, χωρίς χρήση ισοτιμίας και χωρίς χρήση ελέγχου ροής των δεδομένων, όπως παρουσιάζεται στην εικόνα 5-2.



Εικόνα 5-2. Στιγμιότυπο από τις ρυθμίσεις που πρέπει να γίνουν στο Hyper Terminal.



Εικόνα 5-3. Στιγμιότυπο λειτουργίας με το Hyper Terminal

5.3.2 Έλεγχος του I²C Controller μέσω LabVIEW

5.3.2.1 Εισαγωγή

Για την επίτευξη της επικοινωνίας του προγράμματος με τον μC , μέσω της σειριακής θύρας του υπολογιστή, αναπτύχθηκε ένα “πρωτόκολλο” επικοινωνίας για την επιλογή μιας από τις παρακάτω λειτουργίες του σειριακού πρωτοκόλλου επικοινωνίας I²C:

- Διευθυνσιοδότηση (Addressing)
- Εγγραφή (Write)
- Ανάγνωση (Read)
- Τερματισμός (Stop)

Οι εντολές που δίνονται στον μC συντάσσονται χρησιμοποιώντας ως πρόθεμα το πρώτο γράμμα της αγγλικής ονομασίας της αντίστοιχης λειτουργίας, ακολουθούμενο (όταν απαιτείται), από τον αριθμό προς αποστολή σε δεκαεξαδική μορφή. Συγκεκριμένα, η σύνταξη των εντολών είναι η εξής:

- Διευθυνσιοδότηση: AXX, όπου XX το byte διευθυνσιοδότησης π.χ. A3F.
- Εγγραφή: WXX, όπου XX το byte προς εγγραφή π.χ. W1E.
- Ανάγνωση: R. Επιστρέφεται από τον μC το byte που διαβάστηκε.
- Τερματισμός (Stop): S

Ο μC μετά από κάθε εντολή επιστρέφει ένα χαρακτήρα ως επιβεβαίωση (εκτός από την περίπτωση της εντολής τερματισμού). Όταν η λειτουργία που επιλέχθηκε από το χρήστη ολοκληρώνεται επιτυχώς, τότε επιστρέφεται ο χαρακτήρας “C” (completed), ενώ στην αντίθετη περίπτωση επιστρέφεται ο χαρακτήρας “E” (error).

Έτσι π.χ. η ακολουθία χαρακτήρων για έναν πλήρη κύκλο εγγραφής της τιμής 59h σε μια συσκευή slave με διεύθυνση “40h” είναι η “A40CW59CS”, ενώ για ένα πλήρη κύκλο ανάγνωσης από μια συσκευή slave με διεύθυνση “41h”, η ακολουθία είναι η “A41CRC4FS”, όπου με έντονα γράμματα διακρίνονται οι χαρακτήρες που επιστρέφει ο μC .

Συνεπώς, ο σκοπός του προγράμματος είναι η υλοποίηση του συγκεκριμένου πρωτοκόλλου επικοινωνίας.



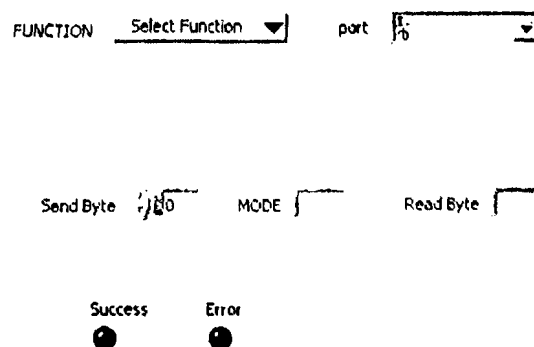
5.3.2.2 Το virtual instrument της συσκευής i2c.vi

Το πρόγραμμα επικοινωνίας αναπτύχθηκε στο περιβάλλον γραφικού προγραμματισμού LabVIEW της εταιρείας National Instruments. Το LabVIEW (Laboratory Virtual Instruments Electronic Workbench) είναι ένα περιβάλλον προγραμματισμού, ιδιαίτερα δημοφιλές στο χώρο των ηλεκτρικών μετρήσεων και βασίζεται στην γλώσσα γραφικού προγραμματισμού “G”. Στο περιβάλλον αυτό, αντί για γραμμές κώδικα, χρησιμοποιούνται έτοιμες συναρτήσεις υπό μορφή αντικειμένων και η διασύνδεση μεταξύ τους επιτυγχάνεται με τη βοήθεια αγωγών, με αποτέλεσμα το πρόγραμμα να μοιάζει περισσότερο σαν ένα διάγραμμα βαθμίδων. Τα προγράμματα καλούνται “εικονικά όργανα” (Virtual Instruments ή VI). [11]

Στην πρόσοψη (front panel) του εικονικού οργάνου (virtual instrument ή vi) διακρίνονται τα εξής πεδία εισαγωγής δεδομένων (σχήμα 5-4):

- “Function”: επιλογή λειτουργίας (διευθυνσιοδότηση, τερματισμός κτλ)
- “Port”: πεδίο επιλογής σειριακής θύρας επικοινωνίας (COM1, COM2 κτλ)
- “Send byte”: byte για διευθυνσιοδότηση ή εγγραφή
- Τα πεδία εξόδου είναι τα εξής:
- “Mode”: Μήνυμα εγγραφής/ανάγνωσης ανάλογα με το bit R/W του byte διευθυνσιοδότησης.
- “Success”: δείκτης επιτυχημένης ανταλλαγής δεδομένων. Ενεργοποιείται όταν επιστραφεί ο χαρακτήρας C από τον μC.
- “Error”: δείκτης σφάλματος. Ενεργοποιείται όταν επιστραφεί ο χαρακτήρας “E” από τον μC ή και σε άλλες περιπτώσεις όπως π.χ. λανθασμένη επιλογή σειριακής θύρας.

Υπάρχουν επίσης και δύο επιπλέον πεδία εξόδου, μη ορατά από το χρήστη, τα οποία χρησιμοποιούνται για αποσφαλμάτωση του προγράμματος (debugging). Τα πεδία αυτά είναι το “error out”, που δηλώνει το είδος και τον κωδικό του σφάλματος, και το “string to μC”, που απεικονίζει την εντολή προς τον μC.



Εικόνα 5-4. Το front panel (πρόσοψη) του εικονικού οργάνου.

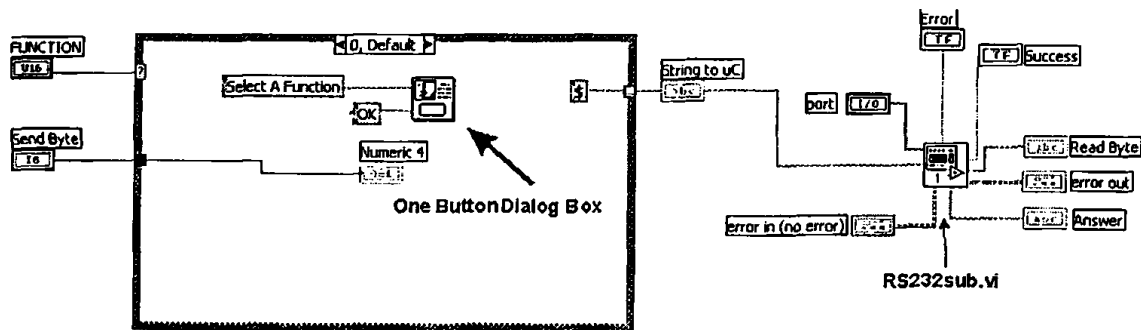
Διάγραμμα Λειτουργίας του VI

Το διάγραμμα λειτουργίας του V, αποτελείται από τις βαθμίδες προετοιμασίας και αποστολής των εντολών προς τον μC. Η βαθμίδα αποστολής των εντολών αποτελεί στην πραγματικότητα ένα sub-VI.

Η λειτουργία της βαθμίδας προετοιμασίας των εντολών, εξαρτάται από την επιλογή του χρήστη στο πεδίο "function". Τα διαγράμματα για καθεμία από τις πέντε πιθανές περιπτώσεις, συμπεριλαμβανομένης και της περίπτωσης που δεν υπάρχει κάποια λειτουργία, απεικονίζονται στα σχήματα 3 έως 8.

Περίπτωση 0 (καμία επιλογή λειτουργίας):

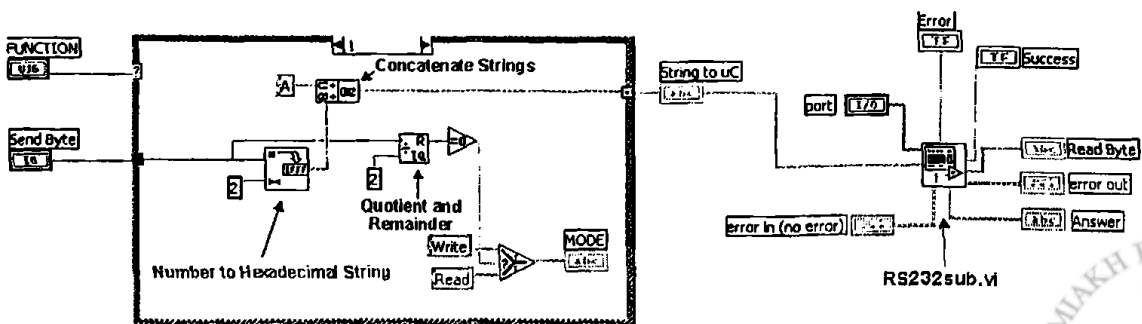
Σε αυτήν την περίπτωση, εμφανίζεται ένα πλαίσιο διαλόγου (dialog box) με το μήνυμα "select a function". Στη βαθμίδα αποστολής, οδηγείται ο χαρακτήρας "\$" τον οποίο αγνοεί ο μC. (σχήμα 5-3)



Σχήμα 5-3. Το διάγραμμα λειτουργίας του Vi για την περίπτωση 0.

Περίπτωση 1 (διευθυνσιοδότηση):

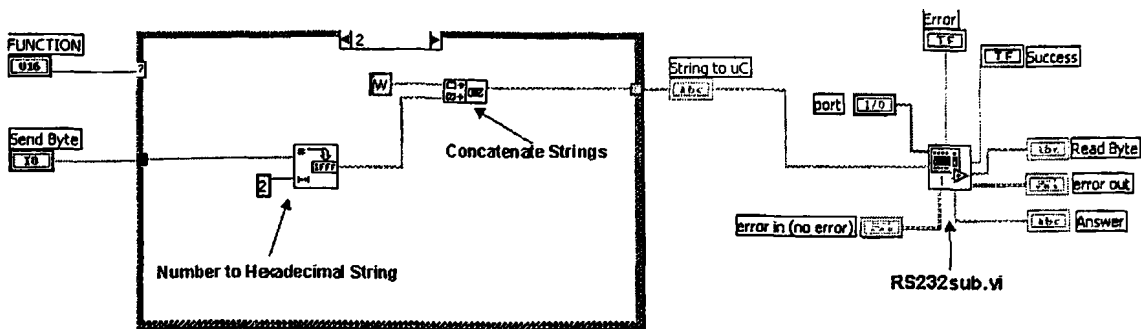
Η δεκαεξαδική τιμή στο πεδίο "send byte", μετατρέπεται σε δυο χαρακτήρες στους οποίους προστίθεται το πρόθεμα "A" και οδηγείται στη βαθμίδα αποστολής. Επιπλέον, ελέγχεται το LSB του byte διευθυνσιοδότησης (R/W) και αναγράφεται το αντίστοιχο μήνυμα στο πεδίο "Mode". (σχήμα 5-4)



Σχήμα 5-4. Το διάγραμμα λειτουργίας του Vi για την περίπτωση 1.

Περίπτωση 2 (εγγραφή):

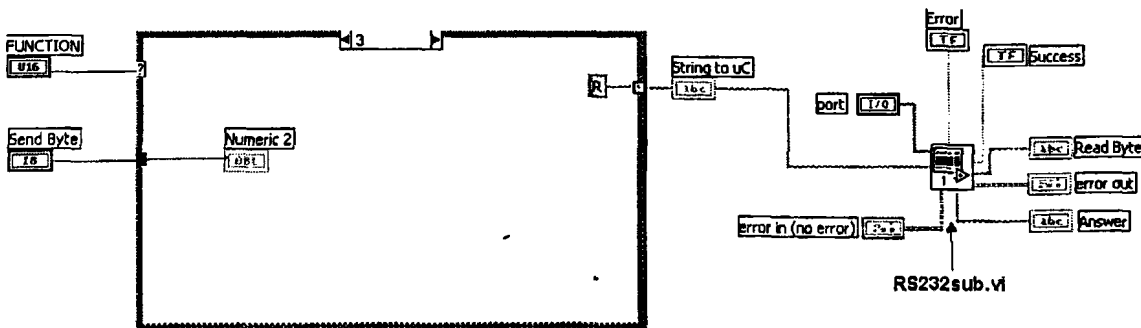
Η δεκαεξαδική τιμή στο πεδίο “send byte”, μετατρέπεται σε δυο χαρακτήρες στους οποίους προστίθεται το πρόθεμα “W” και οδηγείται στη βαθμίδα αποστολής.



Σχήμα 5-5. Το διάγραμμα λειτουργίας του Vi για την περίπτωση 2.

Περίπτωση 3 (ανάγνωση):

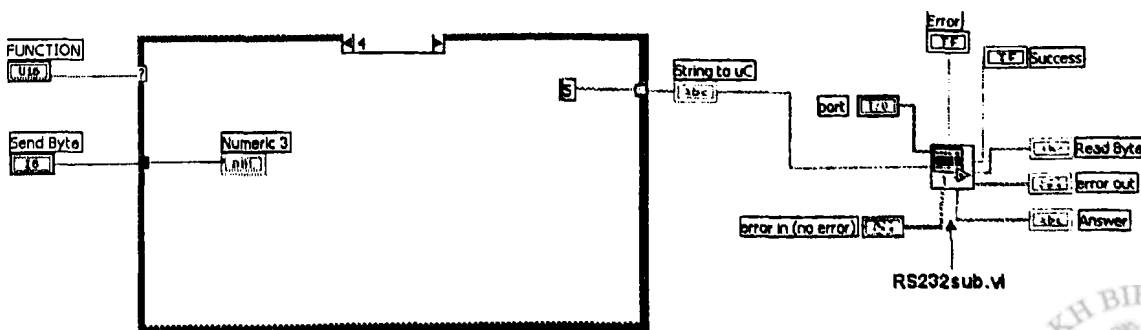
Σε αυτήν την περίπτωση, στη βαθμίδα αποστολής οδηγείται ο χαρακτήρας “R”.



Σχήμα 5-6. Το διάγραμμα λειτουργίας του Vi για την περίπτωση 3.

Περίπτωση 4 (τερματισμός):

Σε αυτήν την περίπτωση, στη βαθμίδα αποστολής οδηγείται ο χαρακτήρας “S”.



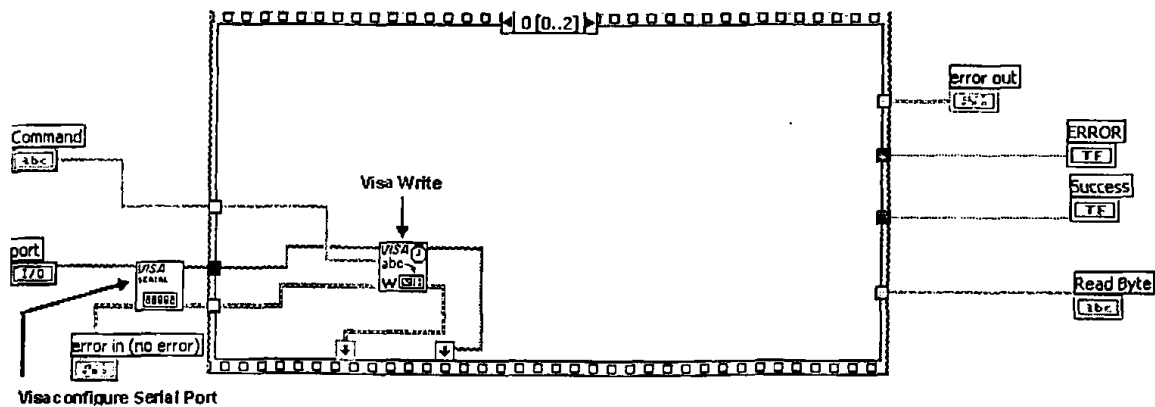
Σχήμα 5-7. Το διάγραμμα λειτουργίας του Vi για την περίπτωση 4.



Βαθμίδα αποστολής των εντολών (RS232sub.vi)

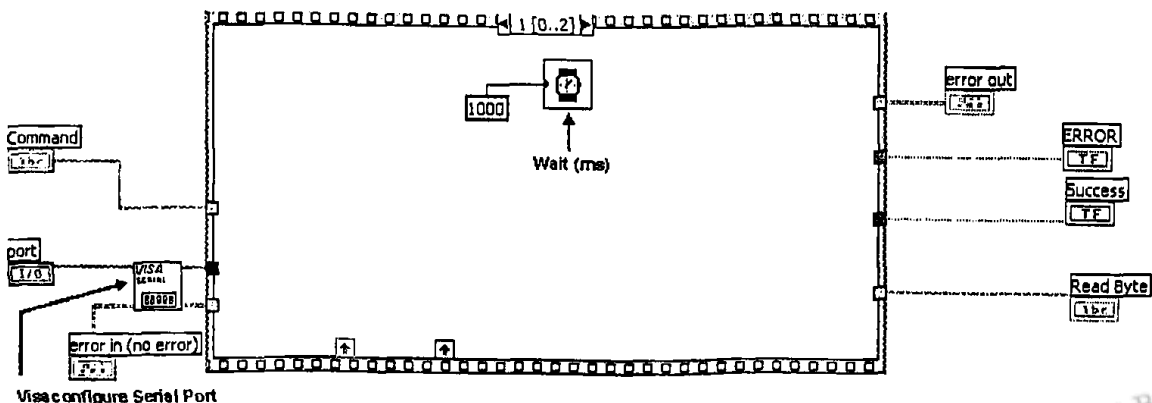
Στη βαθμίδα αποστολής των εντολών, γίνεται ανταλλαγή των δεδομένων μέσω της σειριακής θύρας του υπολογιστή. Η σειριακή θύρα που χρησιμοποιείται για την επικοινωνία, δηλώνεται στο πεδίο "port". Πρώτα γίνεται η αρχικοποίηση των παραμέτρων της θύρας. Στην προκειμένη περίπτωση, επιλέχθηκαν οι εξ ορισμού παράμετροι (ρυθμός μετάδοσης δεδομένων 9600bps, μήκος δεδομένων 8bit, χωρίς ισοτιμία και χωρίς έλεγχο ροής των δεδομένων). Στη συνέχεια εκτελείται μια ακολουθία (που δηλώνεται με το εργαλείο "sequence"), τα στάδια της οποίας είναι τα ακόλουθα:

Βήμα 1: Αποστέλλονται οι χαρακτήρες που εισάγονται από την προηγούμενη βαθμίδα.



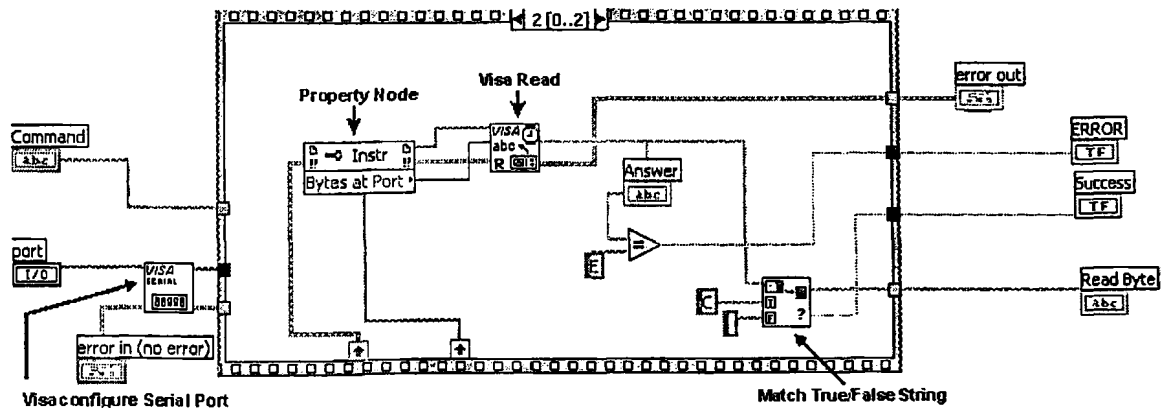
Σχήμα 5-8. Το διάγραμμα λειτουργίας του Vi για το βήμα 1.

Βήμα 2: Αναμονή 1000msec.



Σχήμα 5-9. Το διάγραμμα λειτουργίας του Vi για το βήμα 2.

Βήμα 3: Αρχικά, γίνεται ανάγνωση των εισερχομένων bytes. Στη συνέχεια, ελέγχεται αν έχει επιστραφεί ο χαρακτήρας “E” και αν συμβαίνει κάτι τέτοιο ενεργοποιείται ο δείκτης σφάλματος “Error”. Στην περίπτωση που ο πρώτος χαρακτήρας που λαμβάνεται είναι ο “C”, τότε ενεργοποιείται ο δείκτης επιτυχίας “Success”. Επιπλέον, εμφανίζονται στο πεδίο “read byte” οι υπόλοιποι χαρακτήρες, εκτός του χαρακτήρα “C”. Με αυτόν τον τρόπο, στην περίπτωση ανάγνωσης, εμφανίζονται στο πεδίο “read byte” οι χαρακτήρες που δηλώνουν την τιμή που διαβάστηκε, ενώ στις περιπτώσεις διευθυνσιοδότησης ή εγγραφής, δεν εμφανίζεται τίποτα στο πεδίο “read byte”.



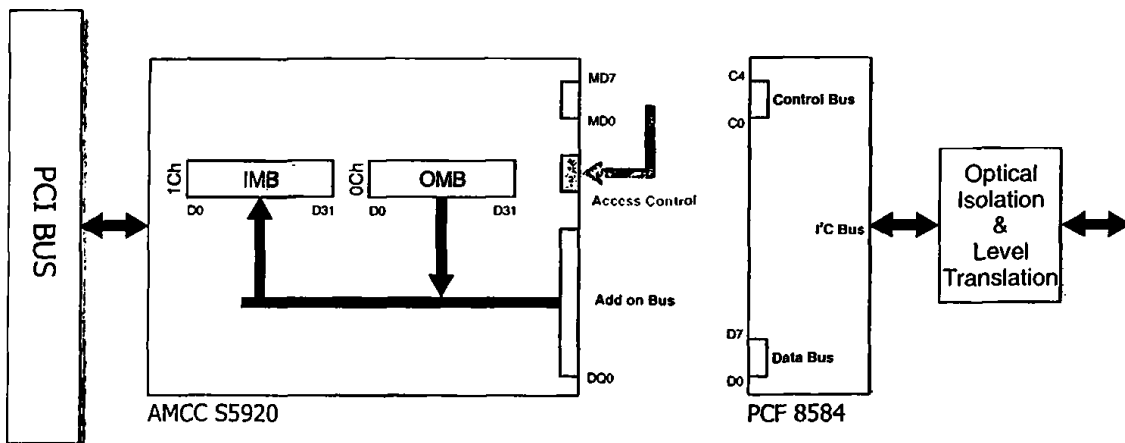
Σχήμα 5-10. Το διάγραμμα λειτουργίας του Vi για το βήμα 3.



6. Κάρτα PCI

6.1 Γενικά

Η κάρτα PCI σχεδιάστηκε με σκοπό να οδηγήσει συσκευές slave που συμβαδίζουν με τη απλή λειτουργία του διαύλου I²C (standard mode). Πρόκειται δηλαδή για μια συσκευή I²C master και μπορεί να παράγει σήμα ρολογιού SCL μέχρι και ~100KHz. Από την πλευρά του PCI η κάρτα αποτελεί μια slave device. Είναι δηλαδή μια κάρτα που δεν μπορεί να ξεκινήσει από μόνη της μια συναλλαγή PCI παρά μόνο αν γίνει αίτηση συναλλαγής από κάποιον master (initiator) του διαύλου PCI. Στην προκειμένη περίπτωση, ο initiator είναι η CPU του Η/Υ που φιλοξενεί την κάρτα.



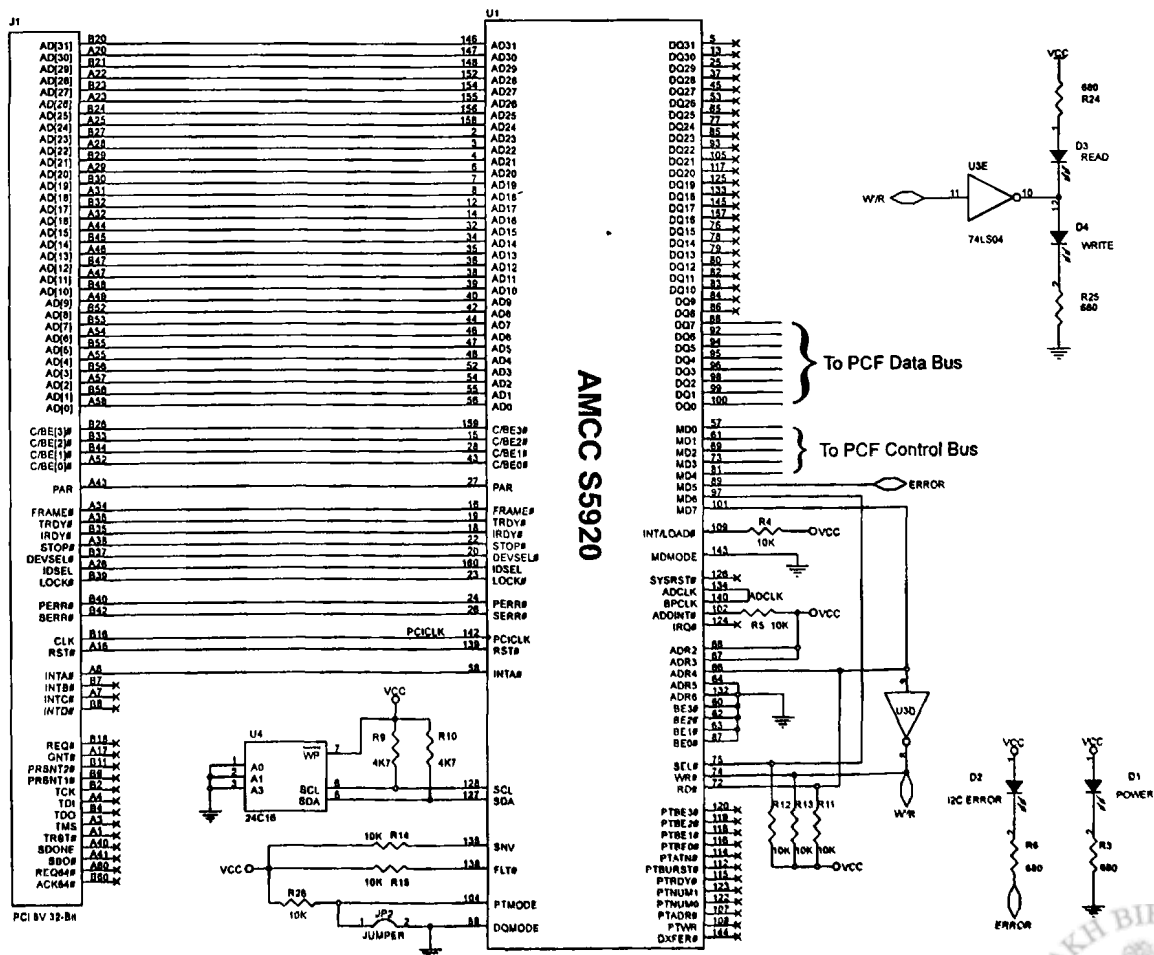
Σχήμα 6-1. Το δομικό διάγραμμα της κάρτας PCI.

Η κάρτα μπορεί να φιλοξενηθεί σε συνδέσμους PCI των 5 V και 32-bit και μπορεί να λειτουργήσει μέχρι τα 33 MHz. Κατά την εκκίνηση του Η/Υ η κάρτα απαιτεί και δεσμεύει πόρους I/O των 128 byte από το σύστημα και για τη λειτουργία της απαιτείται ειδικό λογισμικό οδήγησης ανάλογα με το λειτουργικό σύστημα που είναι εγκατεστημένο στο σύστημα. [12] Η κάρτα μπορεί να χωριστεί σε τρεις βαθμίδες: τη βαθμίδα διασύνδεσης με το δίαυλο PCI, τη βαθμίδα παραγωγής σημάτων I²C και τη βαθμίδα απομόνωσης και προσαρμογής επιπέδων. Στις ενότητες που ακολουθούν θα γίνει ανάλυση των επιμέρους βαθμίδων.

6.2 Βαθμίδα διασύνδεσης με το δίαυλο PCI

Η βαθμίδα διασύνδεσης με το δίαυλο PCI βασίζεται στο ολοκληρωμένο κύκλωμα S5920 που παρουσιάστηκε στο κεφάλαιο 3. Το ολοκληρωμένο S5920 λειτουργεί ως πλήρης διεπαφή για το δίαυλο PCI και παρέχει στο χρήστη ένα δίαυλο 32 bit για την υλοποίηση της επιθυμητής εφαρμογής. Στο σχήμα 6-2 εικονίζεται ένα τμήμα του σχηματικού του κυκλώματος όπου φαίνονται οι απαραίτητες συνδέσεις με την υποδοχή PCI του H/Y.

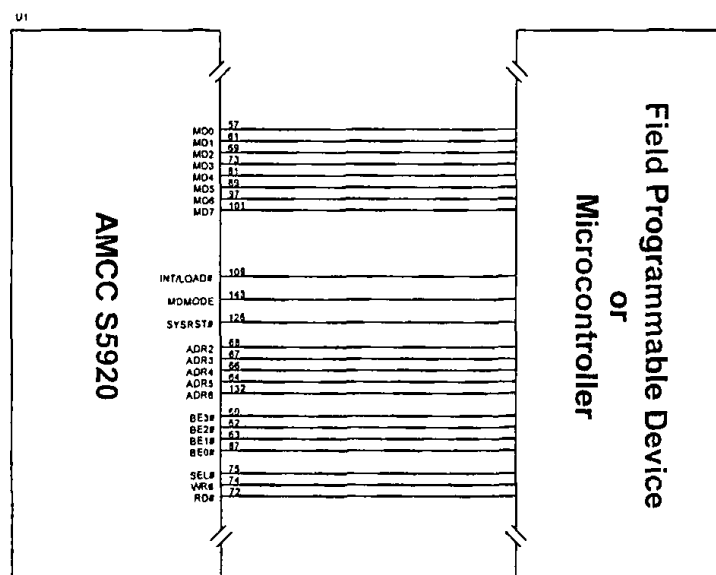
Τα σήματα [AD0:AD32] αποτελούν το τμήμα διευθυνσιοδότησης/δεδομένων του διαύλου PCI είναι δηλαδή το PCI Address-Data bus. Οι ακροδέκτες [C/BE3:C/BE0] ενεργοποιούν μετάδοση των 8-16 ή 32 bit ανάλογα με τις ρυθμίσεις του S5920. Το ολοκληρωμένο U4 είναι μια σειριακή μνήμη EEPROM στην οποία είναι αποθηκευμένες πληροφορίες και ρυθμίσεις για την αρχικοποίηση και τη λειτουργία του S5920.



Σχήμα 6-2. Το σχηματικό διάγραμμα της βαθμίδας διασύνδεσης με το δίαυλο PCI.

Ο ακροδέκτης (104) PTMODE ενεργοποιεί ή απενεργοποιεί τη λειτουργία PASS-THRU και στην παρούσα εφαρμογή διατηρείται σε λογικό '1' (απενεργοποίηση) μέσω ενός αντιστάτη pull-up. Στην παρούσα εφαρμογή χρησιμοποιείται η λειτουργία mailbox διότι δεν υπάρχουν απαιτήσεις για υψηλή ταχύτητα μεταφοράς δεδομένων όπου και θα ήταν απαραίτητη η λειτουργία pass-thru η οποία αναλύθηκε στην παράγραφο 3.2. Επιπλέον με τη χρήση της λειτουργίας mailbox παραμένει αδέσμευτη η λειτουργία Pass-Thru για κάποια άλλη χρήση.

Όπως αναφέρθηκε και στην παράγραφο 3.3.3, το S5920 διαθέτει και έναν επιπλέον δίαυλο επικοινωνίας για την λειτουργία mailbox. Ο δίαυλος mailbox direct [MD7:MD0] οδηγείται απευθείας από το υψηλότερο byte του καταχωρητή PCI-OMB (ή Add-On IMB) [AOMB31:AOMB23]. Γράφοντας δηλαδή μια λέξη των 32 bit στον PCI-OMB και επιλέγοντας μετά από τα pin διευθυνσιοδότησης ADR[6:2] τη διεύθυνση "Ch" (Add-On IMB) τότε είναι εφικτή η ανάγνωση των περιεχομένων του καταχωρητή PCI-OMB και εφόσον έχουμε επιλέξει από τα pin ελέγχου του S5920 να διαβάζουμε τον καταχωρητή αυτόν στο δίαυλο Add-On, τότε τα δεδομένα του υψηλότερου byte του δίαυλου Add-On θα ταυτίζονται με τα δεδομένα του δίαυλου MD. Αν αλλάξουμε τον έλεγχο του S5920 (πχ ανάγνωση του MBEF) τα δεδομένα του PCI-OMB θα παραμείνουν αμετάβλητα στο δίαυλο MD μέχρι ως ότου να γίνει πάλι εγγραφή στον PCI-OMB. Κατά τη γενική επομένως λειτουργία Mailbox του S5920 χρειάζεται ένα ψηφιακό ηλεκτρονικό κύκλωμα από την Add-On πλευρά για να υποστηρίζει τη λειτουργία του. Το κύκλωμα αυτό είναι δυνατό να είναι ένα προγραμματιζόμενο ολοκληρωμένο FPD ή ένας μικροελεγκτής ο οποίος θα πρέπει να ελέγχει τα pin ADR[6:2], BE#[3:0] καθώς και τα pin SEL#, WR# και RD# όπως φαίνεται στο σχήμα 6-3.



Σχήμα 6-3. Πιθανή διασύνδεση του S5920 με ένα προγραμματιζόμενο ολοκληρωμένο κύκλωμα για τη λειτουργία Mailbox.

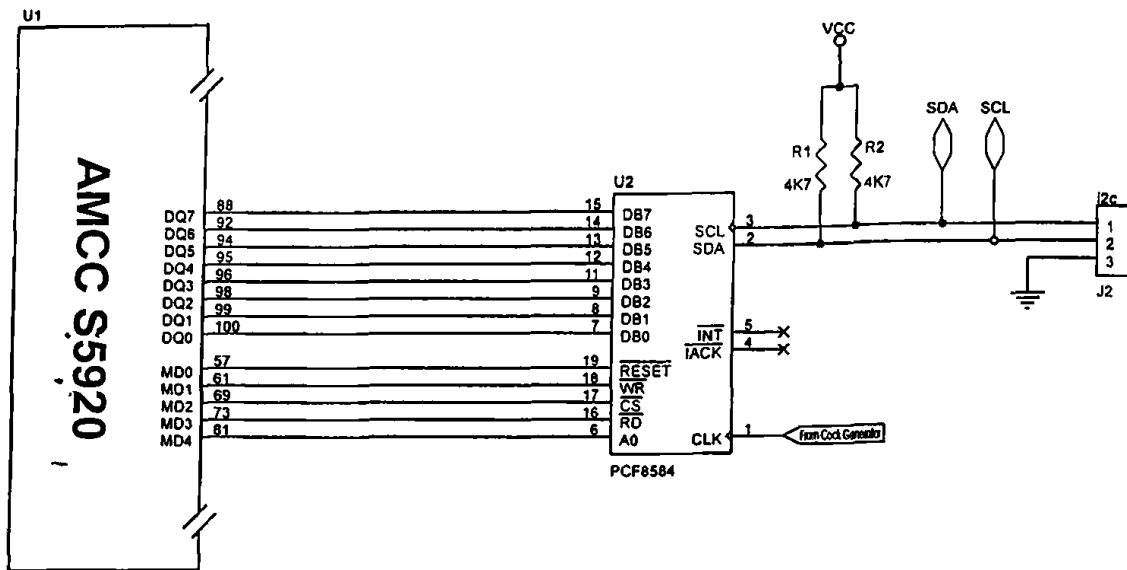
Στην παρούσα εφαρμογή χρησιμοποιήθηκε μια αρκετά απλή λύση η οποία απαλείφει την ύπαρξη ενός πολύπλοκου ηλεκτρονικού κυκλώματος. Εφόσον το S5920 διαθέτει τον δίαυλο Mailbox Direct ο οποίος όπως είδαμε μπορεί και διατηρεί τα δεδομένα στην έξοδο του σταθερά μέχρι ως ότου να γίνει εγγραφή εκ νέου από το δίαυλο PCI, μπορεί να χρησιμοποιηθεί ο ίδιος για τον έλεγχο των παραπάνω pin. Η λύση αυτή περιορίζει το εύρος δεδομένων των καταχωρητών mailbox κατά 2 τουλάχιστον bit, αλλά ταυτόχρονα αναιρεί την ανάγκη ύπαρξης κάποιου πολύπλοκου εξωτερικού κυκλώματος οδήγησης όπως προαναφέραμε. Όπως φαίνεται και από το σχήμα 6-2 χρειάζεται μόνο μια πύλη NOT καθώς και η δέσμευση των 2 πλέον σημαντικών bit του διαύλου MD.

Οι διευθύνσεις των καταχωρητών AIMB και AOMB από την πλευρά Add-On είναι 00011b και 00111b και εφαρμόζονται στον δίαυλο ADR[6:2]. Είναι προφανές ότι διαφέρουν στο τρίτο σημαντικό bit, επομένως κρατώντας σε σταθερή λογική κατάσταση τα υπόλοιπα bits του ADR είναι δυνατή η ανάγνωση και η εγγραφή των καταχωρητών μέσω του διαύλου Add-On με τη χρήση ενός μόνο pin. Όταν το pin ADR4 (66) είναι σε υψηλή λογική κατάσταση τότε έχουμε εγγραφή στον καταχωρητή AOMB, ενώ όταν είναι σε χαμηλή λογική κατάσταση έχουμε ανάγνωση από τον καταχωρητή AIMB.

Τέλος υπάρχουν 4 δίοδοι LED για την οπτική εποπτεία της κάρτας. Η δίοδος D1 δείχνει ότι υπάρχει τάση τροφοδοσίας, η δίοδος D2 χρησιμοποιείται για την ένδειξη λάθους κατά την λειτουργία του διαύλου I²C και οι δίοδοι D3 και D4 δείχνουν την ανάγνωση και εγγραφή στο δίαυλο Add-On.

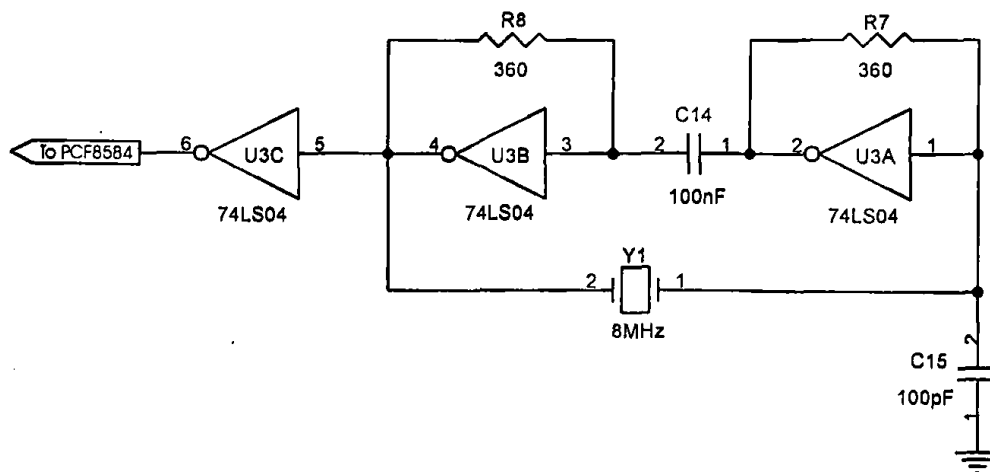
6.3 Βαθμίδα σημάτων I²C

Η βαθμίδα παραγωγής σημάτων I²C (σχήμα 6-4) απαρτίζεται από το ολοκληρωμένο κύκλωμα PCF8584, τους αντιστάτες pull-up R₁, R₂, τον πυκνωτή απόζευξης C13 και τον σύνδεσμο J2. Το ολοκληρωμένο PCF8584 οδηγείται από το PCI bridge S5920 με τη χρήση 13 ακροδεκτών. Από αυτούς οι 8 [DB7:DB0] είναι αμφίδρομοι και οδηγούνται από το λιγότερο σημαντικό byte του διαύλου Add-On, ενώ οι υπόλοιποι είναι ακροδέκτες εξόδου και οδηγούνται από τα σήματα 0 έως 4 του διαύλου Mailbox Direct. Η έξοδος της βαθμίδας I²C οδηγείται στον σύνδεσμο J2 και στη συνέχεια στην επόμενη βαθμίδα, τη βαθμίδα απομόνωσης και προσαρμογής δεδομένων.



Σχήμα 6-4. Το σχηματικό διάγραμμα της βαθμίδας παραγωγής σημάτων I²C.

Για τη λειτουργία του PCF8584 χρειάζεται παλμός ρολογιού συχνότητας 8 MHz, ο οποίος παράγεται από το δικτύωμα των πυλών U3A, U3B, U3C, του κρυστάλλου Y1, και του πυκνωτή C14 όπως φαίνεται στο σχήμα 6-5.



Σχήμα 6-5. Το σχηματικό διάγραμμα της βαθμίδας παραγωγής σήματος ρολογιού 8MHz.



6.4 Βαθμίδα οπτικής απομόνωσης και προσαρμογής επιπέδων

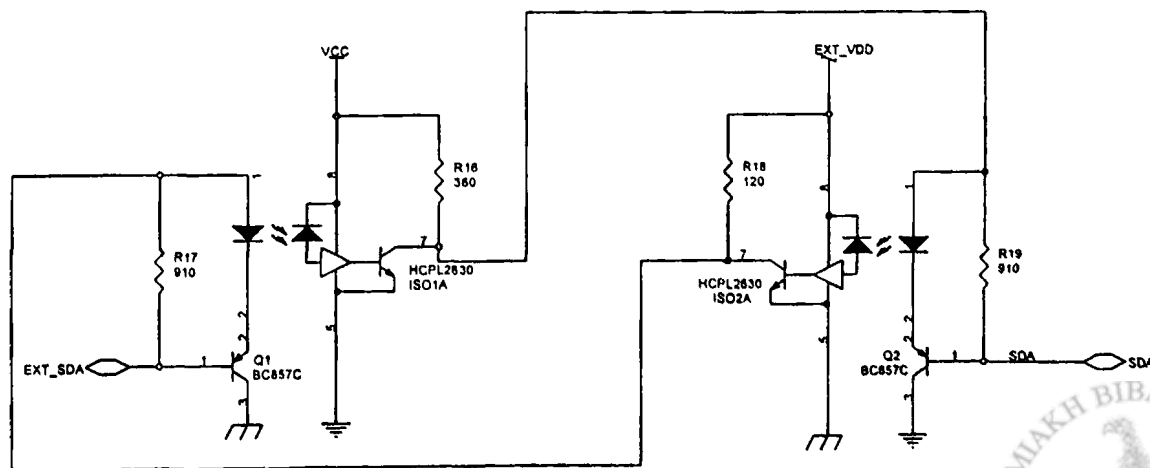
Με τη συνεχή εξέλιξη της τεχνολογίας των ψηφιακών κυκλωμάτων κατασκευάζονται ολοκληρωμένα κυκλώματα τα οποία τροφοδοτούνται με μικρότερη τάση από τη συνηθισμένη μέχρι τώρα τάση τροφοδοσίας των 5 V. Στην κατηγορία των κυκλωμάτων αυτών υπάρχουν και ολοκληρωμένα κυκλώματα που ελέγχονται από το δίαυλο I²C (κάρτες SIM κινητών τηλεφώνων, κάρτες smart, κ.ά.).

Στην περίπτωση κατά την οποία η έξοδος της κάρτας PCI πρέπει να οδηγήσει ένα ολοκληρωμένο που τροφοδοτείται με τάση μικρότερη των 5 V απαιτείται να προηγηθεί μια βαθμίδα προσαρμογής επιπέδων. Στη συνέχεια αναλύονται δύο διατάξεις προσαρμογής επιπέδων. Η πρώτη από αυτές παρέχει και οπτική απομόνωση επειδή είναι βασισμένη σε οπτοζεύκτες. Αμφότερες οι διατάξεις υλοποιήθηκαν στην κάρτα PCI και η αποτίμηση τους έδειξε ότι λειτουργούν κανονικά.

6.4.1 Οπτική απομόνωση και προσαρμογή επιπέδων με χρήση οπτοζευκτών.

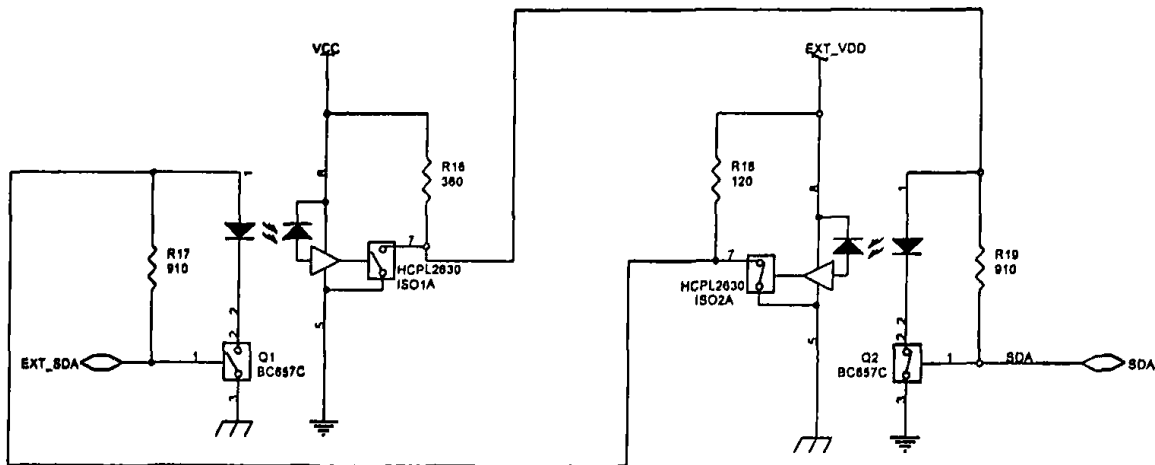
Η βαθμίδα αυτή περιέχει δύο πανομοιότυπες διατάξεις, μία για τη γραμμή SDA και μία για τη γραμμή SCL. Και οι δύο διατάξεις απομόνωσης είναι αμφίδρομες. Λόγω της ομοιότητάς τους, θα αναλυθεί η μία από αυτές. Το πλήρες σχηματικό της κάρτας παρατίθεται στο παράρτημα Β.

Οι έξοδοι του ολοκληρωμένου PCF8584 (σχήμα 6-6) οδηγούνται στις επαφές SDA και SCL, ενώ οι έξοδοι της κάρτας στις οποίες θα συνδεθούν οι εξωτερικές συσκευές στον απομονωμένο πλέον δίαυλο I²C είναι οι επαφές EXT_SDA και EXT_SCL. Η εξωτερική τάση τροφοδοσίας παρέχεται στις επαφές EXT_VDD και EXT_GND και για τις δεδομένες τιμές των αντιστάσεων R₁₈ και R₂₂ είναι 2.5 Volt.



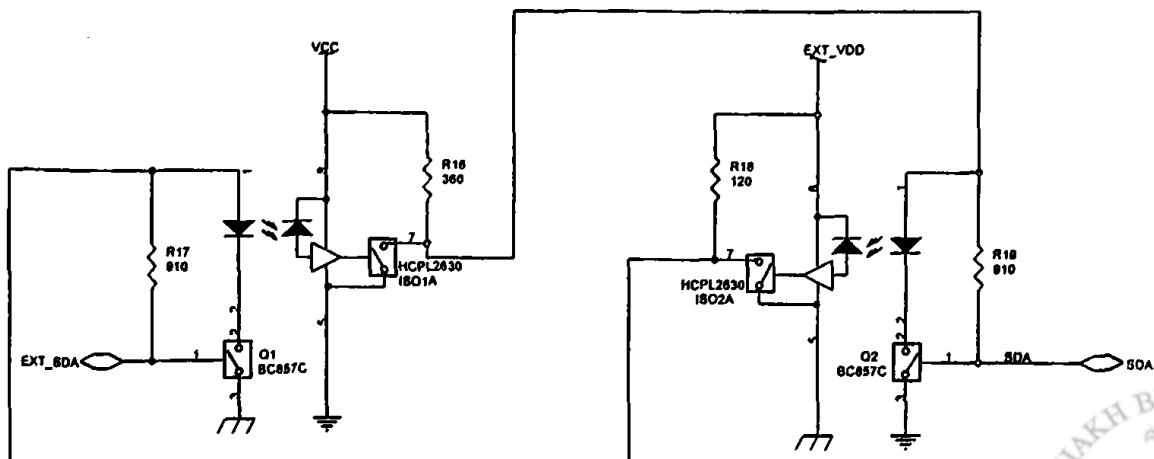
Σχήμα 6-6. Το σχηματικό διάγραμμα της βαθμίδας οπτικής απομόνωσης.

Όταν το ολοκληρωμένο PCF8584 θέτει τη γραμμή SDA σε χαμηλή λογική κατάσταση, ο ηλεκτρονικός διακόπτης Q2 άγει και φωτοβολεί η δίοδος LED που βρίσκεται στο εσωτερικό του ολοκληρωμένου κυκλώματος HCPS2630. Ως αποτέλεσμα της φωτοβολίας άγει και ο εσωτερικός ηλεκτρονικός διακόπτης του οπτοζεύκτη και η γραμμή EXT_SDA έρχεται σε χαμηλή λογική κατάσταση μέσω του αντιστάτη R17, ο οποίος λειτουργεί ως pull-down (Σχήμα 6-7). Ταυτόχρονα αποκόπτεται και το τρανζίστορ Q1 (δεν υπάρχει η απαραίτητη πόλωση) και αποφεύγεται η επιστροφή σήματος στη γραμμή SDA.



Σχήμα 6-7. Η μεταγωγή του κυκλώματος όταν η είσοδος SDA είναι σε λογικό '0'.

Όταν η γραμμή SDA είναι σε υψηλή λογική κατάσταση τότε η εσωτερική LED δεν φωτοβολεί, το εσωτερικό τρανζίστορ παραμένει σε αποκοπή και η γραμμή EXT_SDA έρχεται σε υψηλή λογική κατάσταση μέσω του σύνθετου αντιστάτη pull-up (R17+R18).



Σχήμα 6-8. Η μεταγωγή του κυκλώματος όταν η είσοδος SDA είναι σε λογικό '1'.

Κατά ανάλογο τρόπο γίνεται και η μεταγωγή στο κύκλωμα όταν αυτό οδηγείται από τη γραμμή EXT_SDA

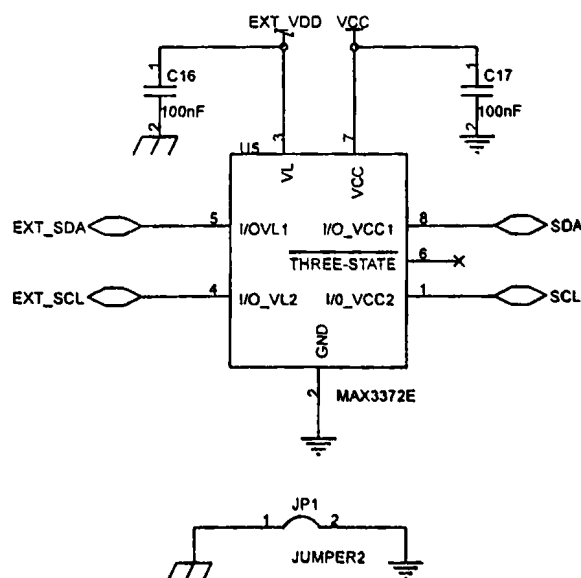
Είναι προφανές ότι στην περίπτωση που είναι επιθυμητό οι γραμμές EXT_SDA και EXT_SCL να μεταγούν μεταξύ 0 και 5V θα πρέπει οι αντιστάτες R18 και R22 να έχουν ίδιες τιμές με τους αντιστάτες R16 Και R20 (βλέπε συνολικό σχηματικό διάγραμμα στο παράρτημα X). Ο τύπος με τον οποίο μπορούν να υπολογιστούν οι τιμές των αντιστατών αυτών είναι οι ακόλουθες:

$$R_x = \frac{(EXT_VDD - EXT_GND) - V_{LED}}{I_{LED}}$$

όπου : V_{LED} είναι περίπου 1,2 Volt και I_{LED} 10 mA, για να υπάρχει αρκετή φωτεινότητα στο LED, ώστε να οδηγηθεί στον κόρο το εσωτερικό τρανζίστορ του οπτοζεύκτη.

6.4.2 Προσαρμογή επιπέδων με χρήση του OK MAX3372

Η βαθμίδα του σχήματος 6-9 υλοποιείται εναλλακτικά της βαθμίδας των οπτοζευκτών. Βασίζεται στο ολοκληρωμένο κύκλωμα MAX3372 ή το MAX3373 της εταιρίας Maxim.[13] Και οι δύο τύποι του ολοκληρωμένου αυτού αποτελούν προσαρμογείς επιπέδων ειδικούς για σειριακές μεταδόσεις και βρίσκουν εφαρμογές κυρίως σε κυκλώματα ανάγνωσης / εγγραφής έξυπνων καρτών, καθώς και καρτών SIM για κινητά τηλέφωνα.

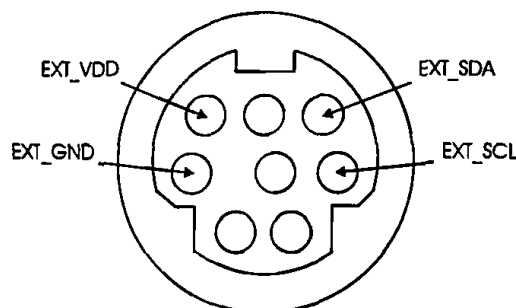


Σχήμα 6-9. Το σχηματικό διάγραμμα της βαθμίδας προσαρμογής επιπέδων.

Οι δύο τύποι ολοκληρωμένου έχουν διαφορά στην ταχύτητα μεταφοράς δεδομένων. Το ολοκληρωμένο MAX3372 υποστηρίζει μεταφορά δεδομένων με μέγιστο ρυθμό 230 Kbps, ενώ το MAX3373 με ρυθμό 8 Mbps. Μια επιπλέον διαφορά είναι ότι στην περίπτωση του MAX3372 χρειάζονται εξωτερικοί αντιστάτες pull-up από την πλευρά EXT_SDA/ EXT_SCL. Στην περίπτωση αυτή, οι αντιστάτες pull-up μπορούν να τοποθετηθούν στη θέση των αντιστατών R17 και R21 με τους αντιστάτες R18 και R22 βραχυκυκλωμένους (βλέπε τοποθέτηση των εξαρτημάτων παράγραφο 6.6). Η τιμή των αντιστατών R17 και R21 στην περίπτωση αυτή είναι 4,7 ΚΩ (σχήμα 6-12).

6.5 Διάταξη των ακίδων του συνδέσμου της κάρτας.

Η διάταξη των ακίδων του συνδέσμου εξόδου της κάρτας εικονίζεται στο σχήμα 6-10



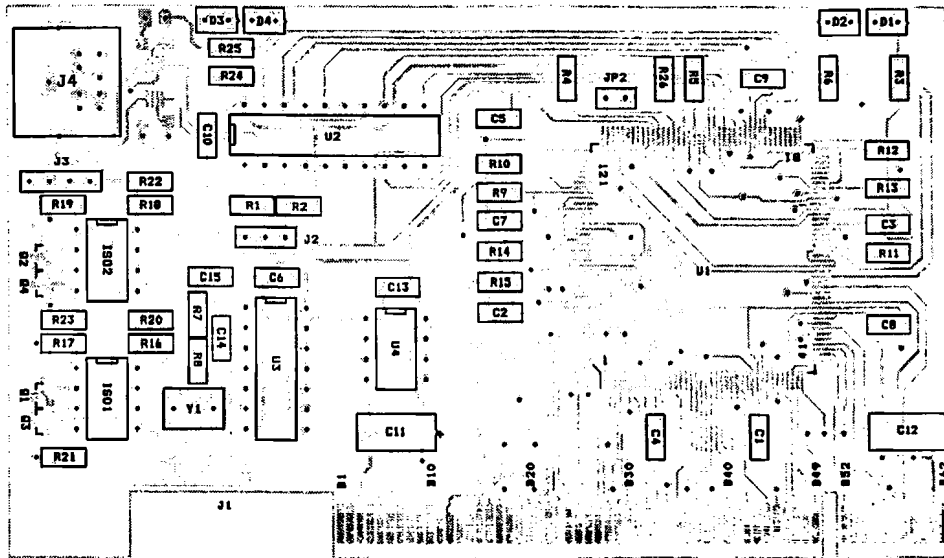
Σχήμα 6-10. Η διάταξη των ακίδων του συνδέσμου εξόδου I²C της κάρτας.

6.6 Τοποθέτηση των εξαρτημάτων στην κάρτα PCI

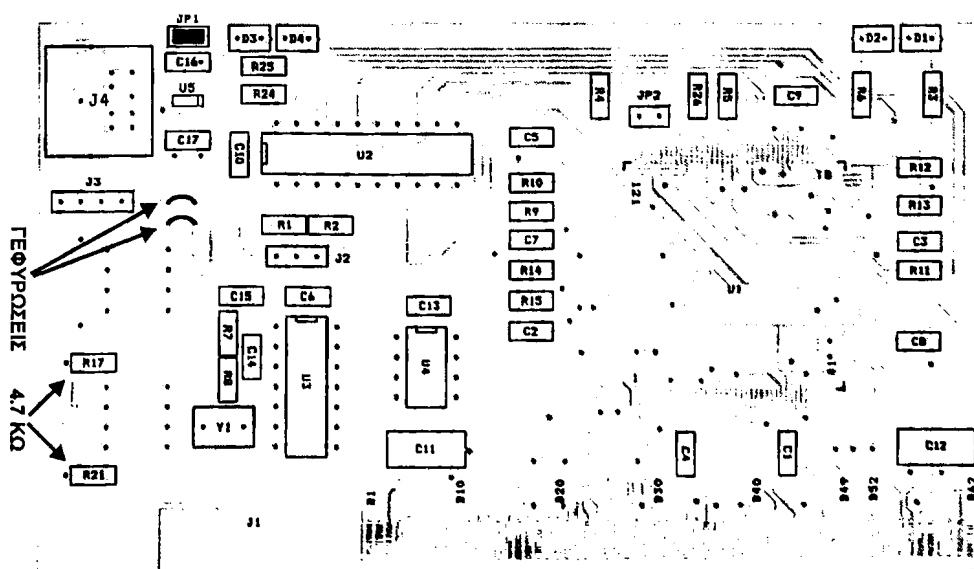
Όπως προαναφέρθηκε η κάρτα PCI υποστηρίζει 2 κυκλώματα προσαρμογής επιπέδων τα οποία υλοποιούνται εναλλακτικά το ένα του άλλου. Για το λόγο αυτό ιδιαίτερη προσοχή θα πρέπει να δοθεί κατά την τοποθέτηση των εξαρτημάτων πάνω στην κάρτα PCI. Συγκεκριμένα προκύπτουν 3 παραλλαγές της βασικής κάρτας. Η πρώτη από αυτές περιλαμβάνει τους οπτοζεύκτες, (σχήμα 6-11) και απουσιάζουν τα εξαρτήματα U5, C16, C17 και JP1. Η δεύτερη παραλλαγή βασίζεται στο ολοκληρωμένο U5 (MAX3372) και στα εξαρτήματα C16, C17, R17, R21 με

γεφυρωμένους τους αντιστάτες R18, R22. Σημειώνεται ότι ο βραχυκυκλωτήρας JP1 υπάρχει και είναι βραχυκυκλωμένος (σχήμα 6-12).

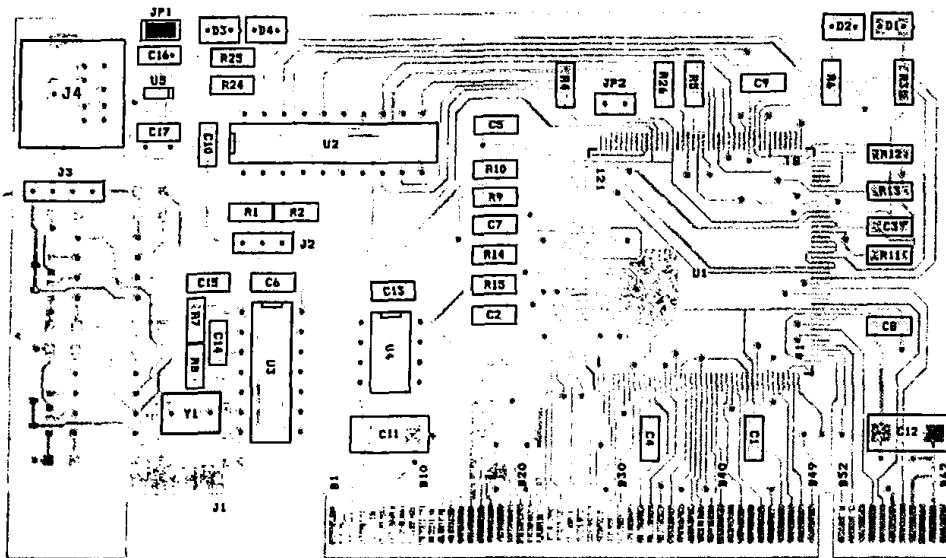
Η Τρίτη παραλλαγή βασίζεται στο ολοκληρωμένο U5 (MAX3372) και στα εξαρτήματα C16, C17. Και εδώ υπάρχει ο βραχυκυκλωτήρας JP1 και είναι βραχυκυκλωμένος (σχήματα 6-13 και 6-14).



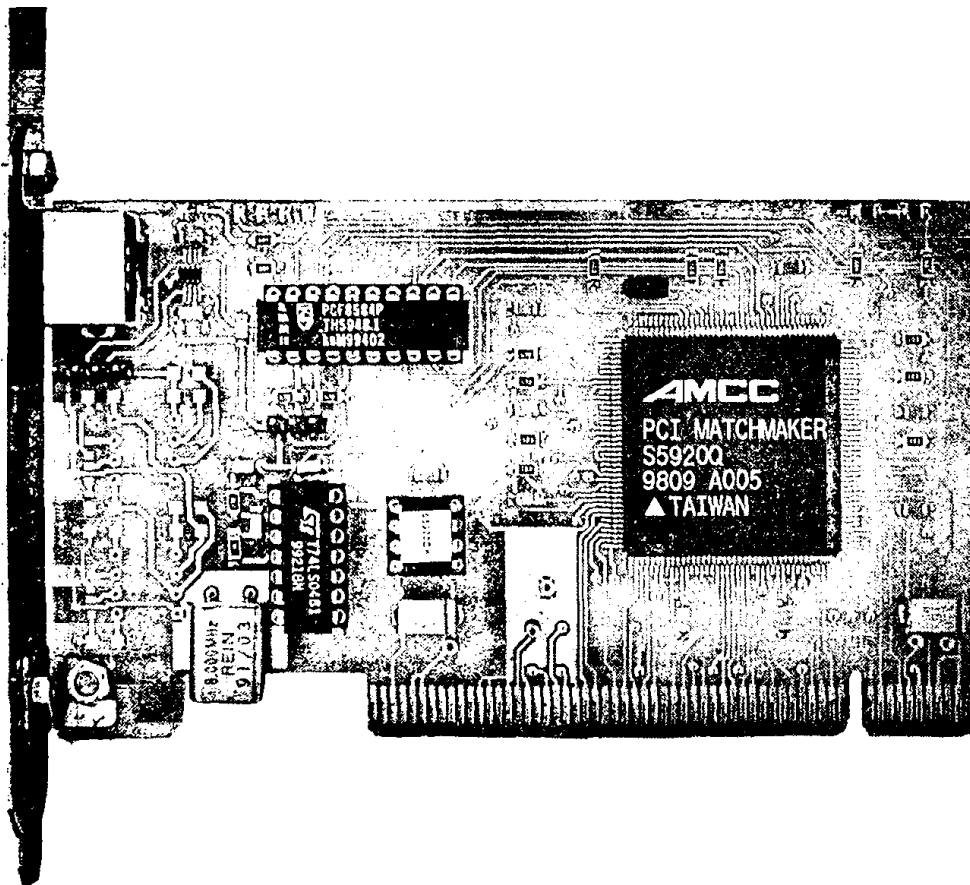
Σχήμα 6-11. Τοπογραφικό τοποθέτησης εξαρτημάτων που περιλαμβάνει προσαρμογή επιπέδων με οπτοζεύκτες.



Σχήμα 6-12. Τοπογραφικό τοποθέτησης εξαρτημάτων που περιλαμβάνει προσαρμογή επιπέδων με το MAX3372.



Σχήμα 6-13. Τοπογραφικό τοποθέτησης εξαρτημάτων που περιλαμβάνει προσαρμογή επιπέδων με το MAX3373.



Σχήμα 6-14. Φωτογραφία της κάρτας PCI σε σχεδόν πραγματικό μέγεθος.

6.7 Οδήγηση της κάρτας PCI

Όπως προαναφέραμε για τη λειτουργία της κάρτας χρειάζεται ειδικό λογισμικό οδήγησης (device driver) το οποίο πρέπει να αναπτυχθεί ειδικά για το λειτουργικό σύστημα (πχ Windows, Linux, VxWorks) κάτω από το οποίο θα λειτουργήσει η κάρτα PCI. Σκοπός του λογισμικού οδήγησης είναι :

- Η ικανοποίηση των απαιτήσεων σε πόρους της κάρτας PCI
- Η υλοποίηση γενικών συναρτήσεων για την ανάγνωση και εγγραφή των καταχωρητών λειτουργίας IMB και OMB αντίστοιχα.

Μαζί με την υλοποίηση των παραπάνω γενικών συναρτήσεων, ο χρήστης θα πρέπει να προγραμματίσει και την εκτέλεση μιας ακολουθίας εντολών για τη σωστή λειτουργία της κάρτας. Με άλλα λόγια το πρόγραμμα οδήγησης θα πρέπει να αποθηκεύει κατάλληλες τιμές στον καταχωρητή OMB και να διαβάζει όποτε αυτό είναι αναγκαίο τον καταχωρητή IMB. Για να γίνει αντιληπτός ο ρόλος που περιγράφουμε ακολουθεί ένα παράδειγμα εγγραφής στον δίαυλο I²C μέσω της κάρτας PCI.

Έστω ότι είναι επιθυμητό να αποσταλούν δεδομένα στο δίαυλο I²C σε μια συσκευή slave με διεύθυνση "40h". Πρέπει να ακολουθηθεί η αλληλουχία των καταστάσεων που παραθέτονται στον πίνακα 6-1. Ως βοήθημα για την κατανόηση των βημάτων μπορούν χρησιμοποιηθούν και τα λογικά διαγράμματα που παρατέθηκαν στην παράγραφο 4.3. Μετά το πέρας και του βήματος 29 θα πρέπει να γίνει ανάγνωση του καταχωρητή IMB. Η ανάγνωση του IMB πρέπει να επαναληφθεί έως ότου το λιγότερο σημαντικό bit του IMB γίνει ίσο με μηδέν. Το γεγονός αυτό δείχνει ότι ο δίαυλος I²C δεν είναι κατειλημμένος.

Στη συνέχεια πρέπει να εκτελεστούν τα βήματα 31 έως και 45 όπου θα πρέπει να γίνει εκ νέου ανάγνωση του καταχωρητή IMB. Η ανάγνωση του IMB πρέπει να επαναληφθεί έως ότου το σημαντικό bit του χαμηλότερου byte του IMB γίνει ίσο με μηδέν. Το γεγονός αυτό δείχνει ότι η διεύθυνση slave στάλθηκε με επιτυχία στο δίαυλο.

Κατόπιν εκτελούνται τα βήματα 47 έως και 50. Ακολουθεί η ανάγνωση του καταχωρητή για άλλη μια φορά. Αν το bit 3 του καταχωρητή IMB βρίσκεται σε υψηλή λογική κατάσταση σημαίνει ότι δεν επιβεβαιώθηκε η διεύθυνση από κάποια συσκευή slave. Σε μια τέτοια περίπτωση πρέπει να σταλεί μια διαδικασία STOP (βήματα S1 έως S5). Διαφορετικά συνεχίζεται η εκτέλεση των υπόλοιπων βημάτων ξεκινώντας από το βήμα 52.

Σε οποιοδήποτε σημείο της εκτέλεσης μπορεί να γίνει επανεκκίνηση της διαδικασίας. Δεν πρέπει όμως να παραληφθεί η πρώτη εντολή η οποία προκαλεί επανεκκίνηση του ολοκληρωμένου PCF8584.



Αριθμός Εντολής	Περιγραφή Λειτουργίας	AMCC								Λέξη 32 bit που εγγράφεται στον OMB				
		Q021	Q030	Q028	Q026	Q017	Q025	Q024	Q023	Q022	Q021-24	Q020-18	Q019-16	Q018-15
		Α	Β	Γ	Δ	Ε	ΣΤ	Ζ	Η	Θ	Ι	Κ	Λ	Μ
1	Reset PCF8584	0	1	0	1	1	1	1	0	5E	XX	XX	XX	
2	Κύκλος εγγραφής στο PCF 8584. Αποθηκεύει την τιμή 80h στον S1, που σημαίνει ότι το επόμενο byte θα αποθηκευτεί στον S0' και απενεργοποιείται η σειριακή διεπαφή.	0	0	0	1	1	1	1	1	1F	XX	XX	80	
3		0	0	0	1	1	0	1	1	1B	XX	XX	80	
4		0	0	0	1	1	0	0	1	19	XX	XX	80	
5		0	0	0	1	1	0	1	1	1B	XX	XX	80	
6		0	0	0	1	1	1	1	1	1F	XX	XX	80	
7		0	0	0	0	1	1	1	1	0F	XX	XX	55	
8	Κύκλος εγγραφής στο PCF 8584. Αποθηκεύει την τιμή 55h στον S0'. Η τιμή της διεύθυνσης slave είναι AAh.	0	0	0	0	1	0	1	1	0B	XX	XX	55	
9		0	0	0	0	1	0	0	1	09	XX	XX	55	
10		0	0	0	0	1	0	1	1	0B	XX	XX	55	
11		0	0	0	0	1	1	1	1	0F	XX	XX	55	
12	Κύκλος εγγραφής στο PCF 8584. Αποθηκεύει την τιμή A0h στον S1, που σημαίνει ότι το επόμενο byte θα αποθηκευτεί στον S2	0	0	0	1	1	1	1	1	1F	XX	XX	A0	
13		0	0	0	1	1	0	1	1	1B	XX	XX	A0	
14		0	0	0	1	1	0	0	1	19	XX	XX	A0	
15		0	0	0	1	1	0	1	1	1B	XX	XX	A0	
16		0	0	0	1	1	1	1	1	1F	XX	XX	A0	
17	Κύκλος εγγραφής στο PCF 8584. Αποθηκεύει την τιμή 18h στον S2. Το εξωτερικό ρολόδι είναι 8 MHz και το SCL = 90 kHz.	0	0	0	0	1	1	1	1	0F	XX	XX	18	
18		0	0	0	0	1	0	1	1	0B	XX	XX	18	
19		0	0	0	0	1	0	0	1	09	XX	XX	18	
20		0	0	0	0	1	0	1	1	0B	XX	XX	18	
21		0	0	0	0	1	1	1	1	0F	XX	XX	18	
22	Κύκλος εγγραφής στο PCF 8584. Αποθηκεύει την τιμή C1h στον S1. Ενεργοποίηση της σειριακής διεπαφής, θέτει τις γραμμές SDA και SCL σε λογικό '1'	0	0	0	1	1	1	1	1	1F	XX	XX	C1	
23		0	0	0	1	1	0	1	1	1B	XX	XX	C1	
24		0	0	0	1	1	0	0	1	19	XX	XX	C1	
25		0	0	0	1	1	0	1	1	1B	XX	XX	C1	
26		0	0	0	1	1	1	1	1	1F	XX	XX	C1	
27	Κύκλος ανάγνωσης από το PCF 8584.	1	0	0	1	1	1	1	1	9F	XX	XX	C1	
28		1	0	0	1	1	0	1	1	9B	XX	XX	C1	
29		1	0	0	1	0	0	1	1	93	XX	XX	C1	
30	Ανάγνωση του IMB ==> Επανάληψη ανάγνωσης έως ότου IMB[0] = 1 (I²C BUS NOT BUSY)													
31	Τέλος κύκλου ανάγνωσης επιστροφή του PCF8584 σε κατάσταση ετοιμότητας	1	0	0	1	1	0	1	1	9B	XX	XX	C1	
32		1	0	0	1	1	1	1	1	9F	XX	XX	C1	
33	Κύκλος εγγραφής στο PCF 8584. Αποθήκευση της διεύθυνσης slave της συσκευής με την οποία ζητείται συναλλαγή πχ. "40h"	0	0	0	0	1	1	1	1	0F	XX	XX	40	
34		0	0	0	0	1	0	1	1	0B	XX	XX	40	
35		0	0	0	0	1	0	0	1	9	XX	XX	40	
36		0	0	0	0	1	0	1	1	0B	XX	XX	40	
37		0	0	0	0	1	1	1	1	0F	XX	XX	40	
38	Κύκλος εγγραφής στο PCF 8584. Αποθήκευση στον S1 της τιμής C5h παράγεται μια διαδικασία START αποστέλλεται η διεύθυνση Slave + RD στο διαυλο I ² C και παράγεται ο 9ος παλμός ρολογιού με την επιβεβαίωση ACK του Slave	0	0	0	1	1	1	1	1	1F	XX	XX	C5	
39		0	0	0	1	1	0	1	1	1B	XX	XX	C5	
40		0	0	0	1	1	0	0	1	19	XX	XX	C5	
41		0	0	0	1	1	0	1	1	1B	XX	XX	C5	
42		0	0	0	1	1	1	1	1	1F	XX	XX	C5	
43	Κύκλος ανάγνωσης από το PCF 8584.	1	0	0	1	1	1	1	1	9F	XX	XX	C5	
44		1	0	0	1	1	0	1	1	9B	XX	XX	C5	
45		1	0	0	1	0	0	1	1	93	XX	XX	C5	
46	Ανάγνωση του IMB ==> Επανάληψη έως ότου IMB[7] = 0 (Ενδειξη ότι στάληκε η διεύθυνση slave)													
47	Τέλος κύκλου ανάγνωσης επιστροφή του PCF8584 σε κατάσταση ετοιμότητας	1	0	0	1	1	0	1	1	9B	XX	XX	C5	
48		1	0	0	1	1	1	1	1	9F	XX	XX	C5	
49	Νεος κύκλος ανάγνωσης από το PCF 8584.	1	0	0	1	1	0	1	1	9B	XX	XX	C5	
50		1	0	0	1	0	0	1	1	93	XX	XX	C5	
51	Ανάγνωση του IMB ==> Εάν IMB[3]=1 (δεν υπήρξε επιβεβαίωση) τότε αποστολή διαδικασίας STOP*													
52	Κύκλος ανάγνωσης από το PCF 8584.	1	0	0	1	1	0	1	1	9B	XX	XX	C5	
53		1	0	0	1	1	1	1	1	9F	XX	XX	C5	
54		0	0	0	0	1	1	1	1	0F	XX	XX	ΒΥΤΕ	
55		0	0	0	0	1	0	1	1	0B	XX	XX	ΒΥΤΕ	
56		0	0	0	0	1	0	0	1	09	XX	XX	ΒΥΤΕ	
57	Κύκλος εγγραφής στο PCF 8584. Αποστολή του επιθυμητού Byte στη συσκευή slave	0	0	0	0	1	0	1	1	0B	XX	XX	ΒΥΤΕ	
58		0	0	0	0	1	1	1	1	0F	XX	XX	ΒΥΤΕ	

ΤΕΛΟΣ

* Αποστολή διαδικασίας STOP

S1	Κύκλος εγγραφής στο PCF 8584. Αποθηκεύει την τιμή C3h στον S1. Το PCF8584 αποστέλλει μια διαδικασία STOP στο διαυλο I ² C	0	0	0	1	1	1	1	1	1F	XX	XX	C3
S2		0	0	0	1	1	0	1	1	1B	XX	XX	C3
S3		0	0	0	1	1	0	0	1	19	XX	XX	C3
S4		0	0	0	1	1	0	1	1	1B	XX	XX	C3
S5		0	0	0	1	1	1	1	1	1F	XX	XX	C3

ΤΕΛΟΣ

Πίνακας 6-1. Η ακολουθία εντολών για εγγραφή ενός byte στο διαυλο I²C από την κάρτα PCI.

6.8 Δοκιμή της κάρτας PCI σε περιβάλλον DOS.

Η εταιρία AMCC παρέχει ένα αρχείο το οποίο μπορεί να τρέξει σε λειτουργικό DOS και να προσπελάσει τους καταχωρητές λειτουργίας του ολοκληρωμένου S5920. Το αρχείο αυτό ονομάζεται OPR.EXE και παρέχεται για σκοπούς αποσφαλματοποίησης (debugging). Η κλήση του αρχείου συντάσσεται ως :

- OPR.EXE W:REG=VALUE για εγγραφή
Όπου REG είναι ο καταχωρητής που θέλουμε να προσπελάσουμε και VALUE είναι η τιμή που θέλουμε να γράψουμε σε αυτόν, πάντα σε δεκαεξαδικό .
- OPR.EXE για ανάγνωση όλων των καταχωρητών

Αμέσως μετά την εκτέλεση της κλήσης της εφαρμογής εκτελείται ένας έλεγχος του διαύλου PCI του H/Y όπου αναγνωρίζονται όλα τα περιφερειακά που είναι συνδεδεμένα σε αυτόν τα οποία εμφανίζονται στην οθόνη με μορφή καταλόγου όπως φαίνεται παρακάτω. Το πρόγραμμα περιμένει από τον χρήστη να επιλέξει ποια συσκευή θέλει να προσπελάσει.

```
0) VID=8086, DID=7030, SVID=0000, SID=0000, bus=0, INDEX=0 Intel
1) VID=8086, DID=7000, SVID=0000, SID=0000, bus=0, INDEX=0 Intel
2) VID=5333, DID=5631, SVID=0000, SID=0000, bus=0, INDEX=0 Unknown Vendor
3) VID=1036, DID=0000, SVID=0000, SID=0000, bus=0, INDEX=0 Unknown Vendor
4) VID=10e8, DID=5920, SVID=10e8, SID=00ee, bus=0, INDEX=0 AMCC
```

Enter device selection (99 to exit program) =>

Σχήμα 6-15. Στιγμιότυπο από την εκτέλεση της εφαρμογής OPR.EXE.

Εναλλακτικά της όλης διαδικασίας, ο χρήστης μπορεί να επιλέξει, εκ των προτέρων σε ποια συσκευή απευθύνεται αν δηλώσει κατά την κλήση της εφαρμογής ένα από τα στοιχεία της συσκευής που εμφανίζει ο παραπάνω κατάλογος.

Π.χ. OPR.EXE DID=5920 W:OMB=5EFFFFFF

Με τη τελευταία κλήση της εφαρμογής επιλέγεται η συσκευή 4 με Device ID = 5920 που είναι η κάρτα PCI που αναλύεται στη παρούσα εργασία.

Έχοντας ως εργαλείο την εφαρμογή ο χρήστης μπορεί να υλοποιήσει την ακολουθία του πίνακα 6-1 με την μορφή ενός batch file και στη συνέχεια να εκτελέσει μια εγγραφή στο δίαυλο I²C. Η μορφή του αρχείου batch εικονίζεται στο σχήμα 6-15.


```

@ECHO OFF
OPR.EXE DID=5920 W:OMB=5EFFFFFF >null
OPR.EXE DID=5920 W:OMB=1FFFFFF80 >null
OPR.EXE DID=5920 W:OMB=1BFFFF80 >null
OPR.EXE DID=5920 W:OMB=19FFFF80 >null
OPR.EXE DID=5920 W:OMB=1BFFFF80 >null
OPR.EXE DID=5920 W:OMB=1FFFFFF80 >null
OPR.EXE DID=5920 W:OMB=0FFFFFF55 >null
OPR.EXE DID=5920 W:OMB=0BFFFF55 >null
OPR.EXE DID=5920 W:OMB=09FFFF55 >null
OPR.EXE DID=5920 W:OMB=0BFFFF55 >null
OPR.EXE DID=5920 W:OMB=0FFFFFF55 >null
OPR.EXE DID=5920 W:OMB=1FFFFFFA0 >null
OPR.EXE DID=5920 W:OMB=1BFFFFA0 >null
OPR.EXE DID=5920 W:OMB=19FFFFA0 >null
OPR.EXE DID=5920 W:OMB=1BFFFFA0 >null
OPR.EXE DID=5920 W:OMB=1FFFFFFA0 >null
OPR.EXE DID=5920 W:OMB=0FFFFFF18 >null
OPR.EXE DID=5920 W:OMB=0BFFFF18 >null
OPR.EXE DID=5920 W:OMB=09FFFF18 >null
OPR.EXE DID=5920 W:OMB=0BFFFF18 >null
OPR.EXE DID=5920 W:OMB=0FFFFFF18 >null
OPR.EXE DID=5920 W:OMB=1FFFFFFC1 >null
OPR.EXE DID=5920 W:OMB=1BFFFC1 >null
OPR.EXE DID=5920 W:OMB=19FFFC1 >null
OPR.EXE DID=5920 W:OMB=1BFFFC1 >null
OPR.EXE DID=5920 W:OMB=1FFFC1 >null
OPR.EXE DID=5920 W:OMB=9FFFC1 >null
OPR.EXE DID=5920 W:OMB=9BFFFC1 >null
OPR.EXE DID=5920 W:OMB=93FFFC1 >null
OPR.EXE DID=5920
OPR.EXE DID=5920 W:OMB=9BFFFC1 >null
OPR.EXE DID=5920 W:OMB=9FFFC1 >null
OPR.EXE DID=5920 W:OMB=0FFFF40 >null
OPR.EXE DID=5920 W:OMB=0BFFFF40 >null
OPR.EXE DID=5920 W:OMB=09FFFF40 >null
OPR.EXE DID=5920 W:OMB=0BFFFF40 >null
OPR.EXE DID=5920 W:OMB=0FFFF40 >null
OPR.EXE DID=5920 W:OMB=1FFFC5 >null
OPR.EXE DID=5920 W:OMB=1BFFFC5 >null
OPR.EXE DID=5920 W:OMB=19FFFC5 >null
OPR.EXE DID=5920 W:OMB=1BFFFC5 >null
OPR.EXE DID=5920 W:OMB=1FFFC5 >null
OPR.EXE DID=5920 W:OMB=9FFFC5 >null
OPR.EXE DID=5920 W:OMB=9BFFFC5 >null
OPR.EXE DID=5920 W:OMB=93FFFC5 >null
OPR.EXE DID=5920
OPR.EXE DID=5920 W:OMB=9BFFFC5 >null
OPR.EXE DID=5920 W:OMB=9FFFC5 >null
OPR.EXE DID=5920 W:OMB=9BFFFC5 >null
OPR.EXE DID=5920 W:OMB=93FFFC5 >null
OPR.EXE DID=5920
OPR.EXE DID=5920 W:OMB=9BFFFC5 >null
OPR.EXE DID=5920 W:OMB=9FFFC5 >null
OPR.EXE DID=5920 W:OMB=0FFFFAA >null
OPR.EXE DID=5920 W:OMB=0BFFFAA >null
OPR.EXE DID=5920 W:OMB=09FFFAA >null
OPR.EXE DID=5920 W:OMB=0BFFFAA >null
OPR.EXE DID=5920 W:OMB=0FFFFAA >null

```

Σχήμα 6-16. Ο πίνακας 6-1 σε μορφή batch file.



6.9 Αποτίμηση της κάρτας PCI

- Κατασκευή τυπωμένου

Όπως αναφέρθηκε και στην εισαγωγή η ανάπτυξη του τυπωμένου κυκλώματος της κάρτας PCI έγινε εξ' ολοκλήρου στο εργαστήριο χωρίς τη συμβολή κάποιας βιομηχανίας κατασκευής τυπωμένων. Το αποτέλεσμα της κατασκευής του τυπωμένου υπήρξε εξαιρετικά ικανοποιητικό και στα πέντε στάδια κατασκευής του:

- ❖ φωτομεταφορά
- ❖ χημική επεξεργασία (εμφάνιση και απομάκρυνση του χαλκού)
- ❖ κατασκευή vias
- ❖ συναρμολόγηση
- ❖ συγκόλληση εξαρτημάτων.

- Τεχνικά χαρακτηριστικά εξόδου I²C

Συχνότητα SCL = 90 kHz

Ρυθμός αποστολής δεδομένων 90 kbps.

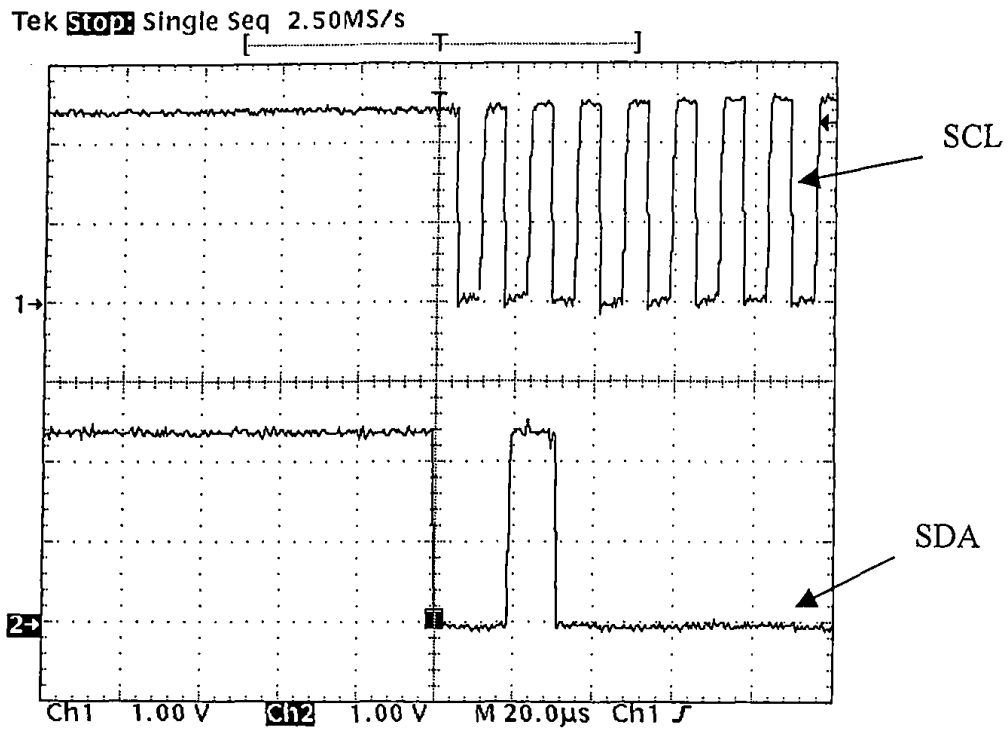
Πλάτος σήματος εισόδου, εξόδου και για τα 2 σήματα SCL, SDA :

- ✓ 2 – 5 volts με χρήση οπτοζευκτών
- ✓ 1,6 – 5 volts με χρήση των MAX3372-73

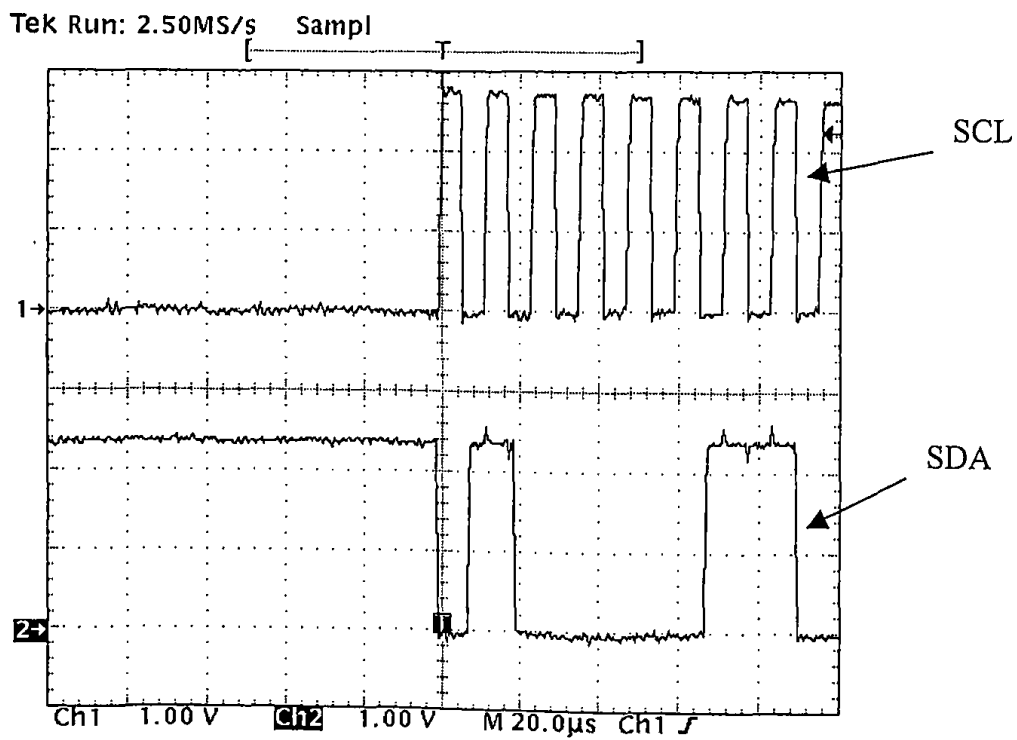
Στην περίπτωση των οπτοζευκτών, η επιλογή των επιπέδων λειτουργίας, γίνεται με την εφαρμογή εξωτερικής τάσης, χρειάζεται όμως αλλαγή των αντιστατών Pull-Up, όπως είδαμε στη παράγραφο 6.4.1. Στην περίπτωση των MAX3372-73, η επιλογή των επιπέδων λειτουργίας, γίνεται με την εφαρμογή εξωτερικής τάσης, χωρίς την αλλαγή κάποιων αντιστατών.

Στις εικόνες 6-1, 6-2 παραθέτονται στιγμιότυπα από τον παλμογράφο κατά την εκτέλεση της ακολουθίας εντολών του πίνακα 6-1. Η συχνότητα του σήματος SCL είναι ~90 kHz ενώ η τάση εξωτερικής τροφοδοσίας EXT_VDD-EXT_GND είναι 2,5 volts. Στην εικόνα 6-1 φαίνονται τα σήματα SCL και SDA κατά την αποστολή της διεύθυνσης slave '40h'. Στην εικόνα 6-2 φαίνονται τα σήματα SCL και SDA κατά την αποστολή του byte '43h'.





Εικόνα 6-1. Η αποστολή της διεύθυνσης '40h' στο δίαυλο I²C.



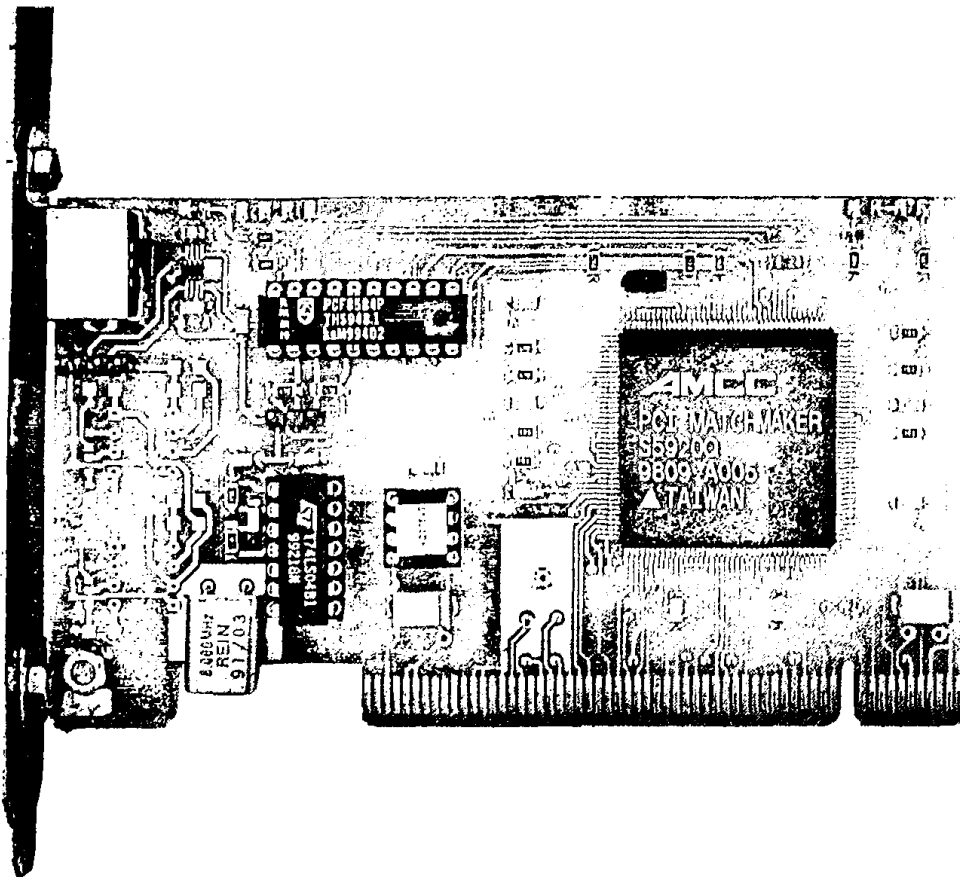
Εικόνα 6-2. Η αποστολή του byte '43h' στο δίαυλο I²C.

Αναφορές – Βιβλιογραφία

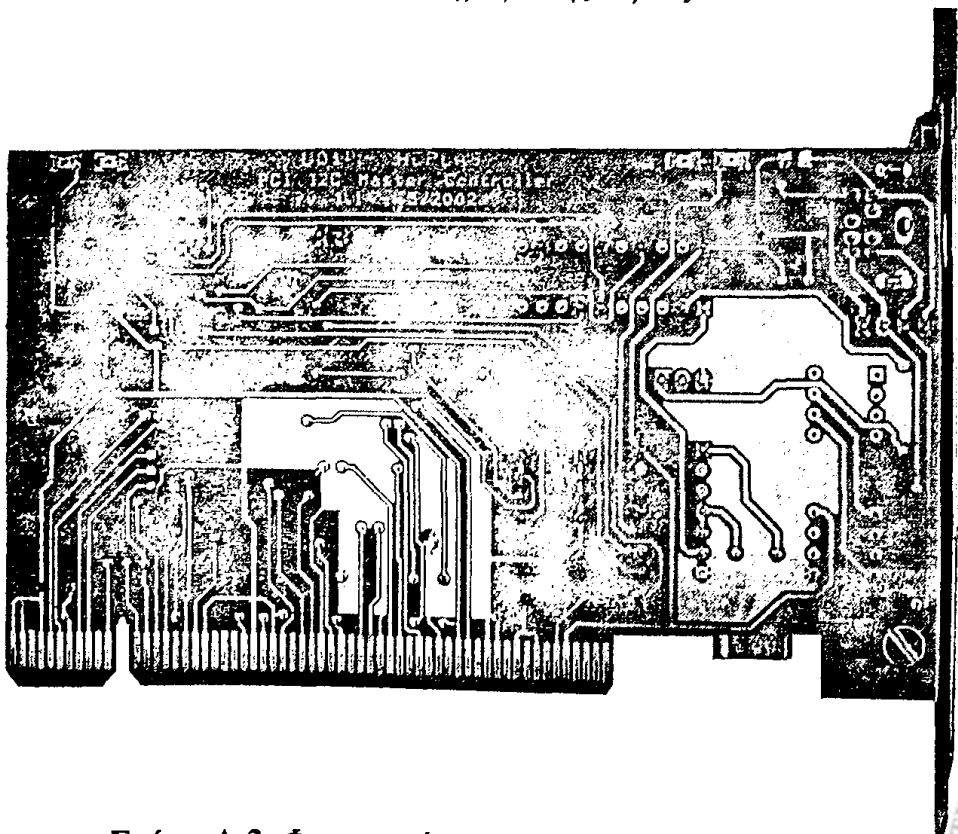
- [1] Philips Semiconductors
The I²C-bus specification version 2.1, 2001
- [2] Ι. Καρράς και Ε. Δράκου
Το Hardware της πληροφορικής
- [3] T. Shanley & D. Anderson
PCI System Architecture 4th edition, Addison Wesley, 1999
- [4] PCI Special Interest Group
PCI Local bus specification revision 2.2, 1998
- [5] A. Rubini
Linux Device Drivers 2nd edition, O'Reilly, 2001
- [6] Applied Micro Circuits Corp.
S5920 PCI Target/Slave Interface Data Book 1st edition, 1997
- [7] Philips Semiconductors
PCF8584 I²C bus Controller Product Specification, 1997
- [8] Motorola Inc.
MC68HC05 Microcontroller Applications Guide rev.3, 1997
- [9] Philips Semiconductors
AN425 Interfacing the PCF8584 I²C-bus controller to 80C51 family microcontrollers, 1994
- [10] Θ. Αμοιρίδης
Εφαρμοσμένη Ηλεκτρονική, Αϊβάζης, 1999
- [11] National Instruments Corp.
LabVIEW 5.1 Function & VI Reference Manual, 1998
- [12] Π. Βηχούδης
Λογισμικό οδήγησης κάρτας PCI διαύλου I²C και μετατροπέας ψηφιακών σημάτων TTL-ECL-NIM-LVDS για πειράματα φυσικής υψηλών ενεργειών, Πανεπιστήμιο Ιωαννίνων, 2002
- [13] Maxim Integrated Products
AN1159 Level Translators for SPI and I2C Bus Signals, 2002

ΠΑΡΑΡΤΗΜ

Φωτογραφίες της κάρτας PCI

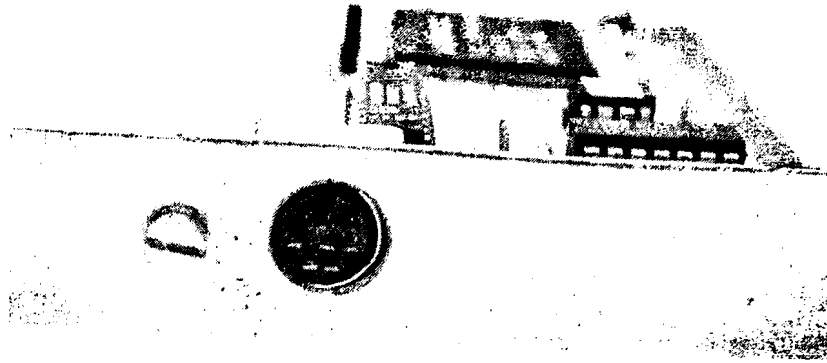


Σχήμα Α-1. Φωτογραφία της κάρτας PCI.

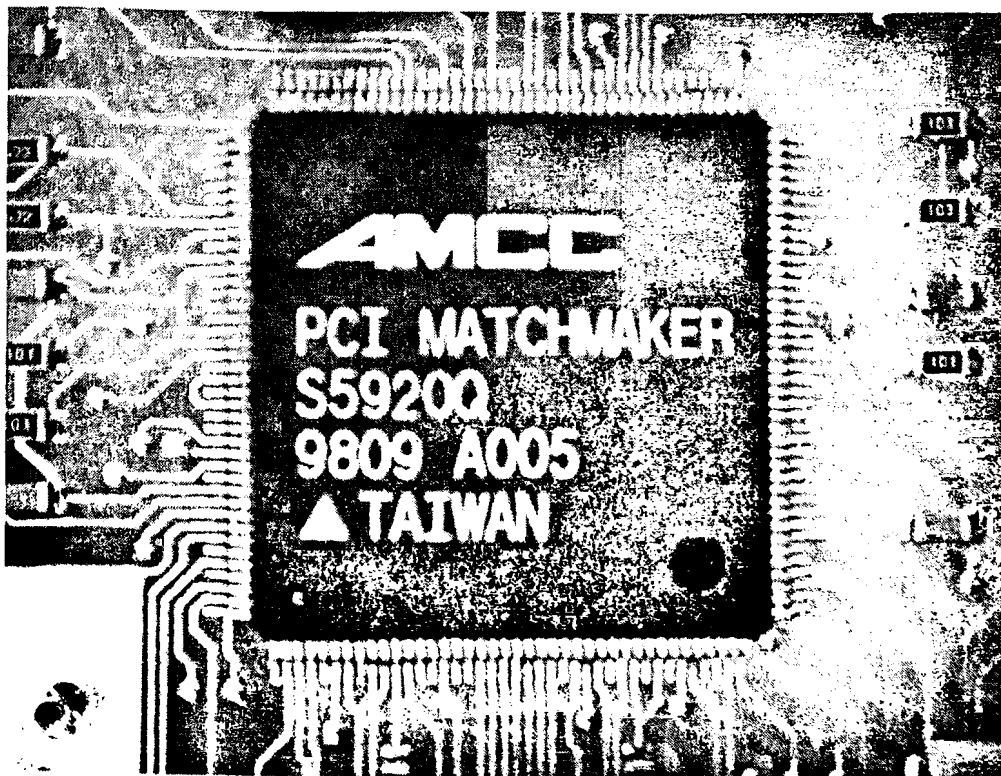


Σχήμα Α-2. Φωτογραφία της κάρτας PCI (πίσω πλευρά).





Σχήμα Α-3. Φωτογραφία της κάρτας PCI, διακρίνεται ο σύνδεσμος εξόδου I/O.



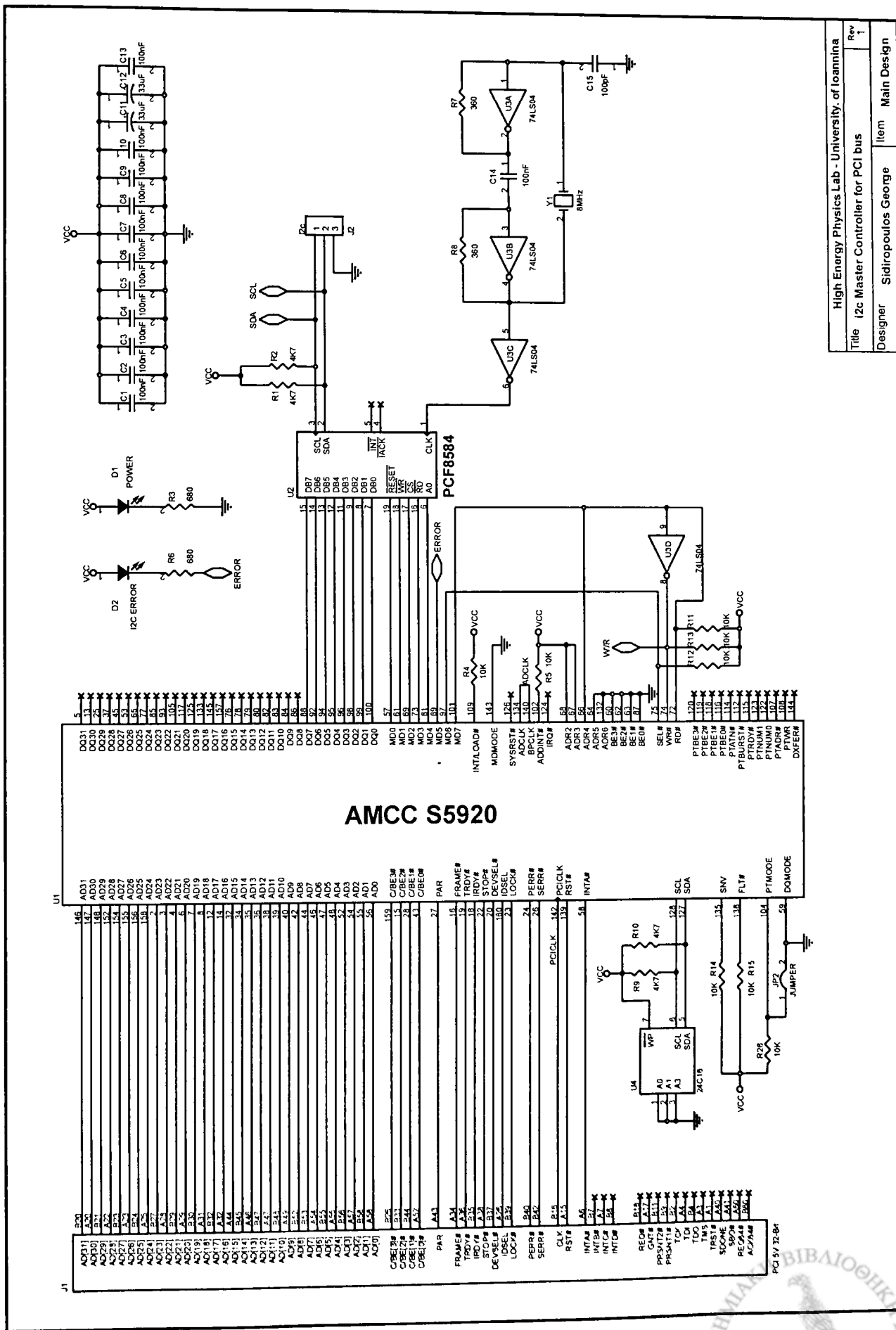
Σχήμα Α-4. Φωτογραφία της κάρτας PCI, διακρίνεται το ολοκληρωμένο διασύνδεσης με το δίαυλο PCI.

ΠΑΡΑΡΤΗΜΑ Β

Ηλεκτρονικά σχέδια και τυπωμένα κυκλώματα

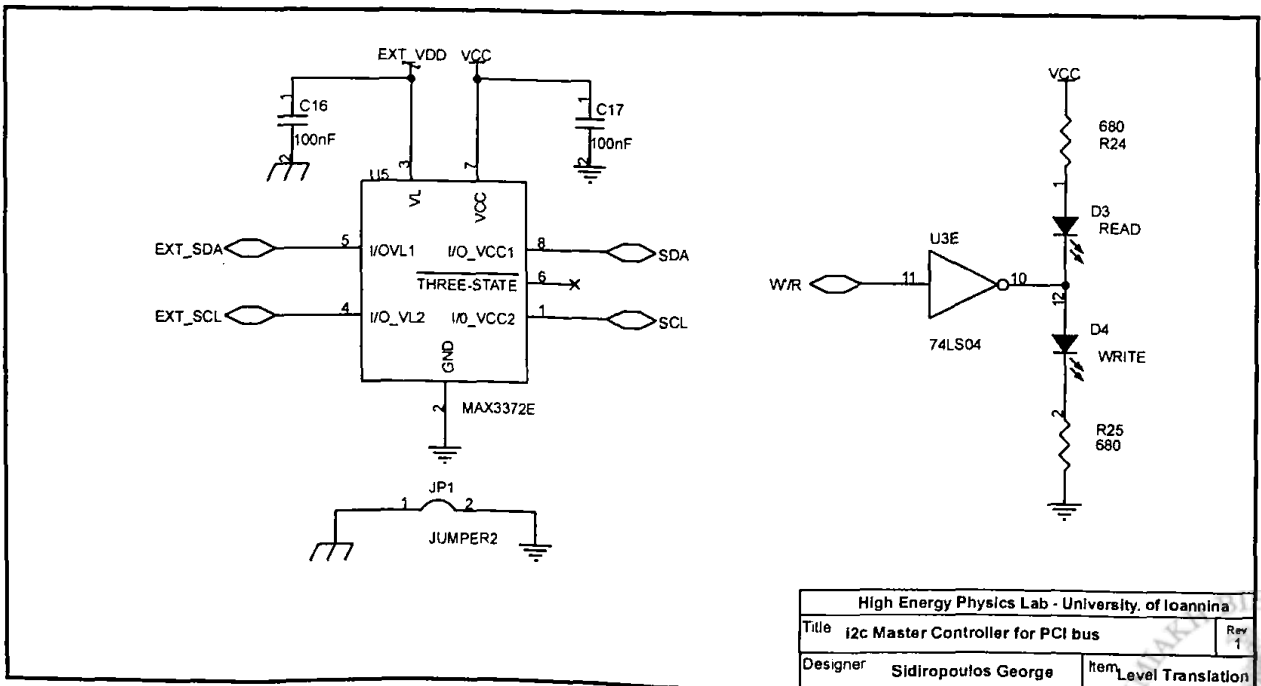
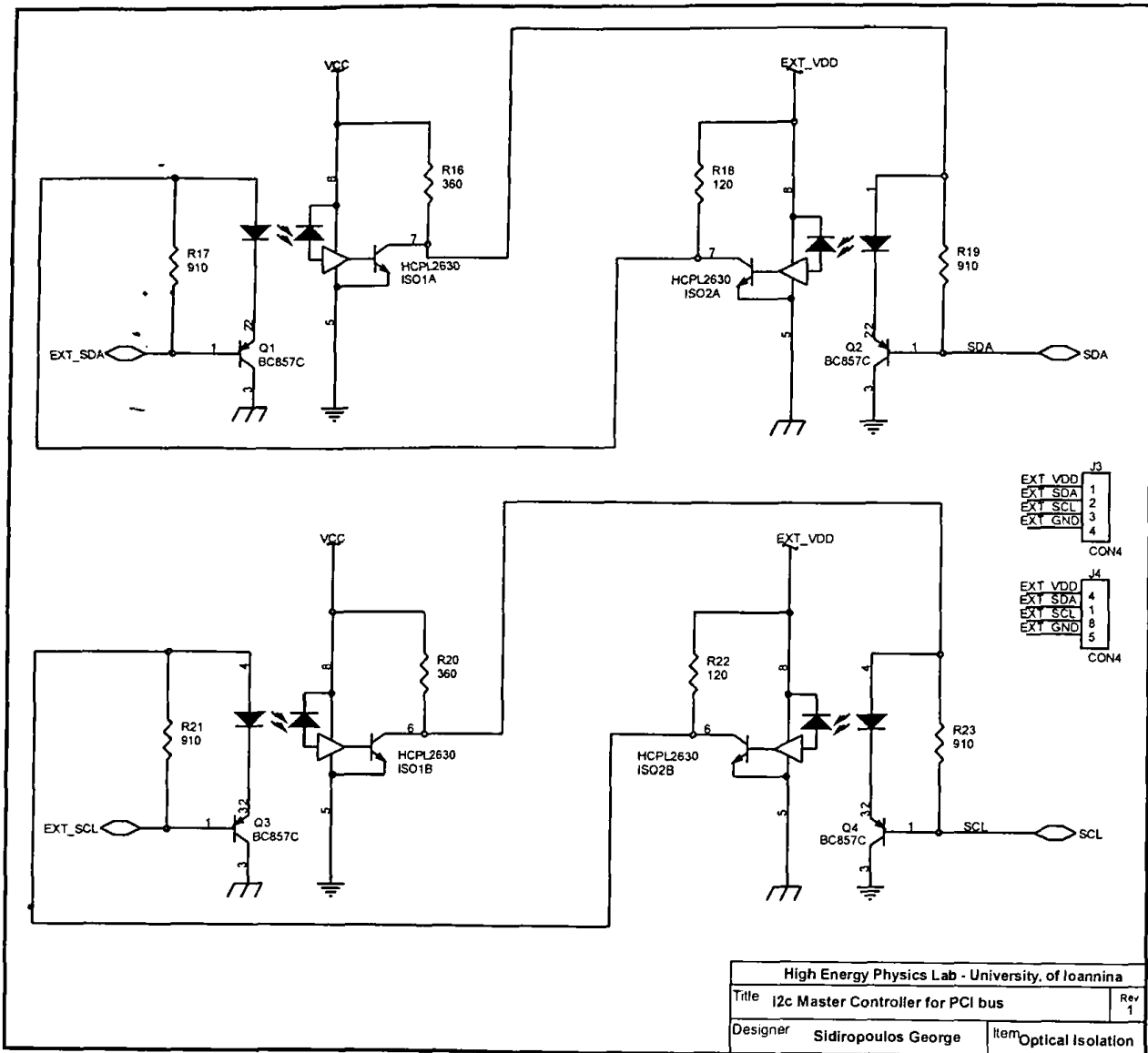
Στο παρόν παράρτημα παραθέτονται:

- ✓ Το ηλεκτρονικό σχέδιο της κάρτας PCI.
- ✓ Το τυπωμένο κύκλωμα της κάρτας PCI.
- ✓ Το τοπογραφικό τοποθέτησης εξαρτημάτων στην κάρτα PCI.
- ✓ Ο κατάλογος των υλικών της κάρτας PCI.
- ✓ Το ηλεκτρονικό σχέδιο του RS-232 I²C Controller.
- ✓ Το τυπωμένο κύκλωμα του RS-232 I²C Controller.
- ✓ Το τοπογραφικό τοποθέτησης εξαρτημάτων στο τυπωμένο του RS-232 I²C Controller.
- ✓ Ο κατάλογος των υλικών του RS-232 I²C Controller.

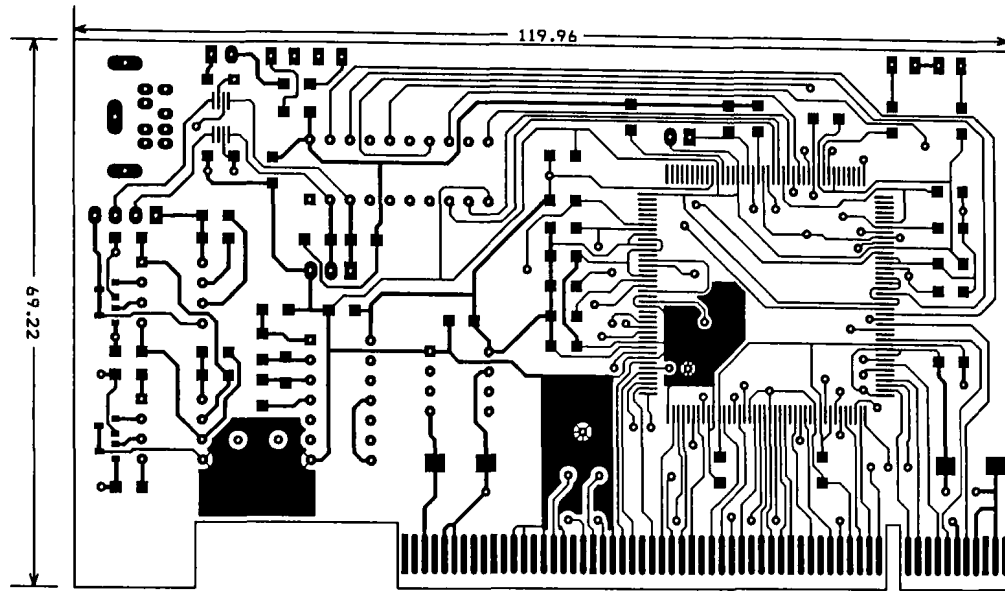


High Energy Physics Lab - University of Ioannina	
Title	i2c Master Controller for PCI bus
Designer	Sidropoulos George
Item	Main Design

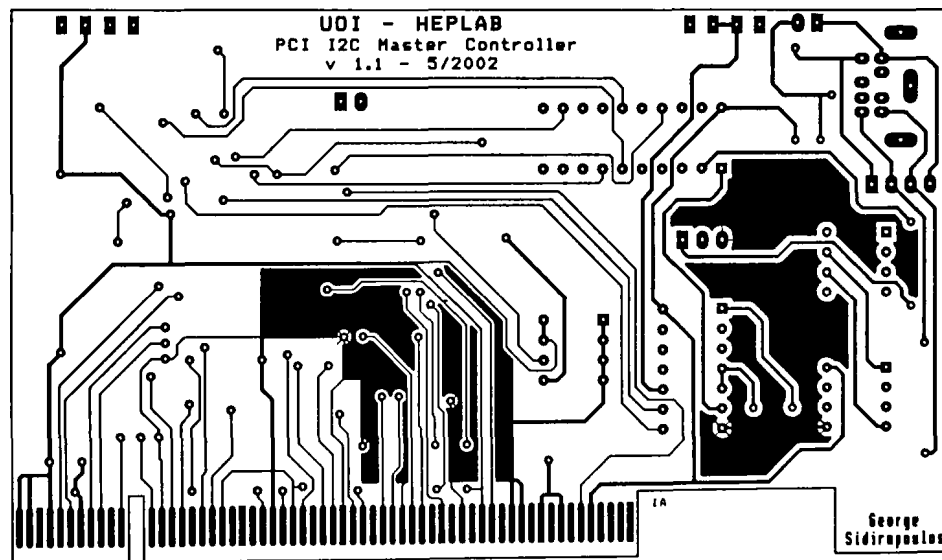
Σχήμα Β-1. Το ηλεκτρονικό σχέδιο της κάρτας PCI.



Σχήμα Β-2. Το ηλεκτρονικό σχέδιο της κάρτας PCI (συνέχεια).



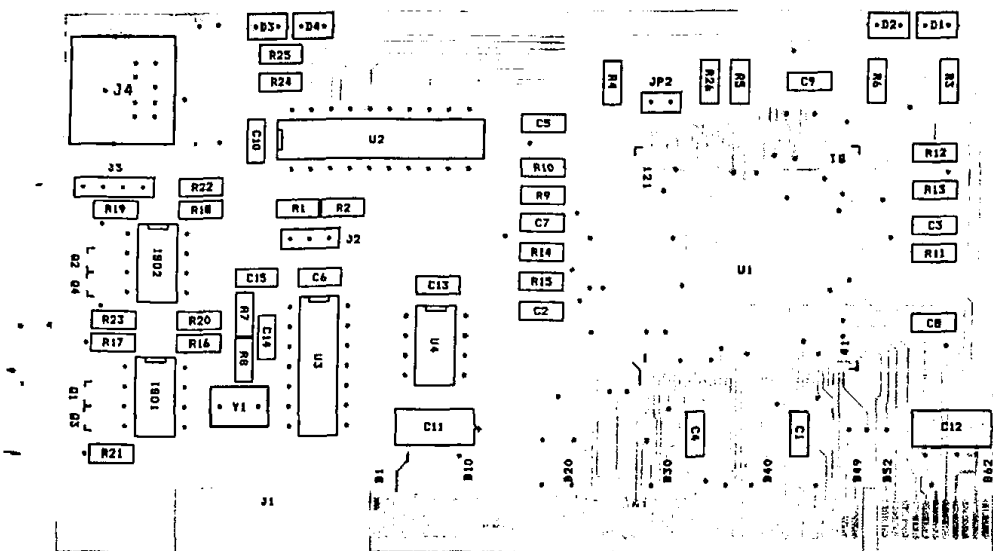
TOP LAYER



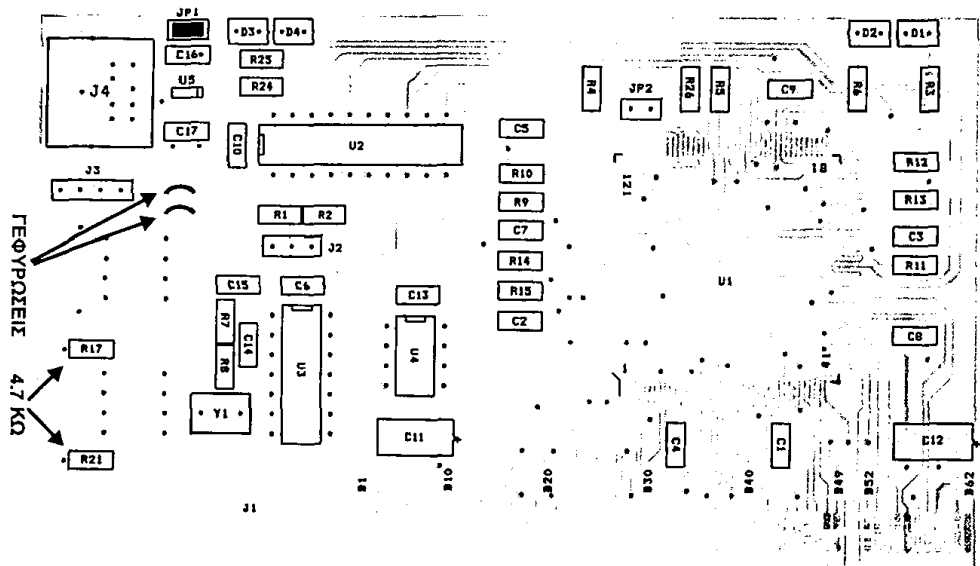
BOTOM LAYER

Σχήμα Β-3. Τα τυπωμένα κυκλώματα της πάνω και κάτω όψης της κάρτας PCI).

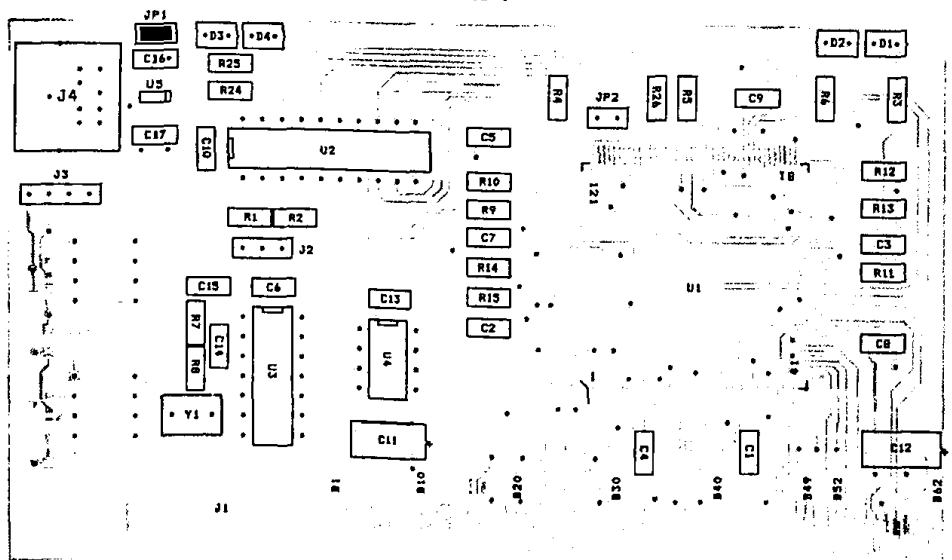




(α)

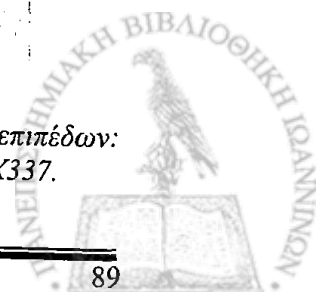


(β)



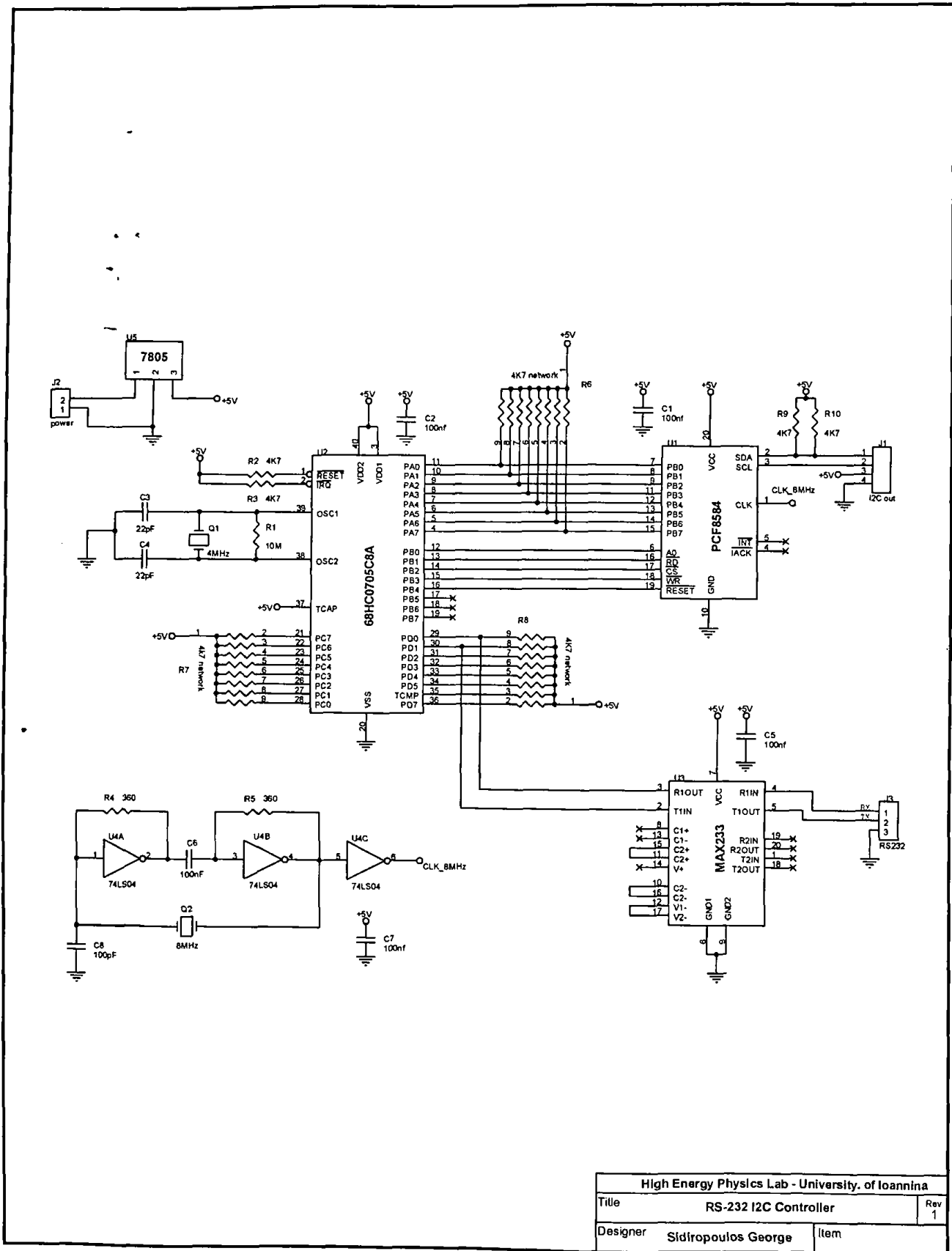
(γ)

Σχήμα Β-4. Τοπογραφικό τοποθέτησης εξαρτημάτων που περιλαμβάνει προσαρμογή επιπέδων:
 α) με οπτοζεύκτες β) με το ολοκληρωμένο MAX3372 γ) με το ολοκληρωμένο MAX337.



Υλικό	Τιμή	Περιγραφή
C1 - C10	100nF	Πυκνωτής πολυεστ. SMD 1206
C11 - C12	33uF/16 V	Πυκνωτής ηλεκτρ. SMD
C13 - C14	100nF	Πυκνωτής πολυεστ. SMD 1206
C15	100pF	Πυκνωτής κεραμ. SMD 1206
C16 - C17	100nF	Πυκνωτής πολυεστ. SMD 1206
D1 - D4	LED	SMD 1206 package
ISO1 - ISO2	HCPL2630	Οπτοζεύκτης
JP1	JUMPER	2 pin-array
JP2	JUMPER	2 pin-array
J2	Σύνδεσμος	3 pin-array
J3	Σύνδεσμος	4 pin-array
J4	Σύνδεσμος	8-pin Mini Din Θηλυκός 90°
Q1 - Q4	BC857C	Transistor SMD
R1 - R2	4K7	Αντιστάτης SMD 1206 package
R3	680	Αντιστάτης SMD 1206 package
R4 - R5	10K	Αντιστάτης SMD 1206 package
R6	680	Αντιστάτης SMD 1206 package
R7 - R8	360	Αντιστάτης SMD 1206 package
R9 - R10	4K7	Αντιστάτης SMD 1206 package
R11 - R15	10K	Αντιστάτης SMD 1206 package
R16	360	Αντιστάτης SMD 1206 package
R17	910	Αντιστάτης SMD 1206 package
R18	120	Αντιστάτης SMD 1206 package
R19	910	Αντιστάτης SMD 1206 package
R20	360	Αντιστάτης SMD 1206 package
R21	910	Αντιστάτης SMD 1206 package
R22	120	Αντιστάτης SMD 1206 package
R23	910	Αντιστάτης SMD 1206 package
R24	680	Αντιστάτης SMD 1206 package
R25	680	Αντιστάτης SMD 1206 package
R26	10K	Αντιστάτης SMD 1206 package
U1	5920	PCI Target Interface
U2	PCF8584	PCI I2C Controller
U3	74LS04	Αντιστροφείας
U4	24C16	Μνήμη EEPROM I2C
U5	MAX3372E	Προσαρμογέας Επιπέδων
Y1	8MHz	Κρύσταλλος

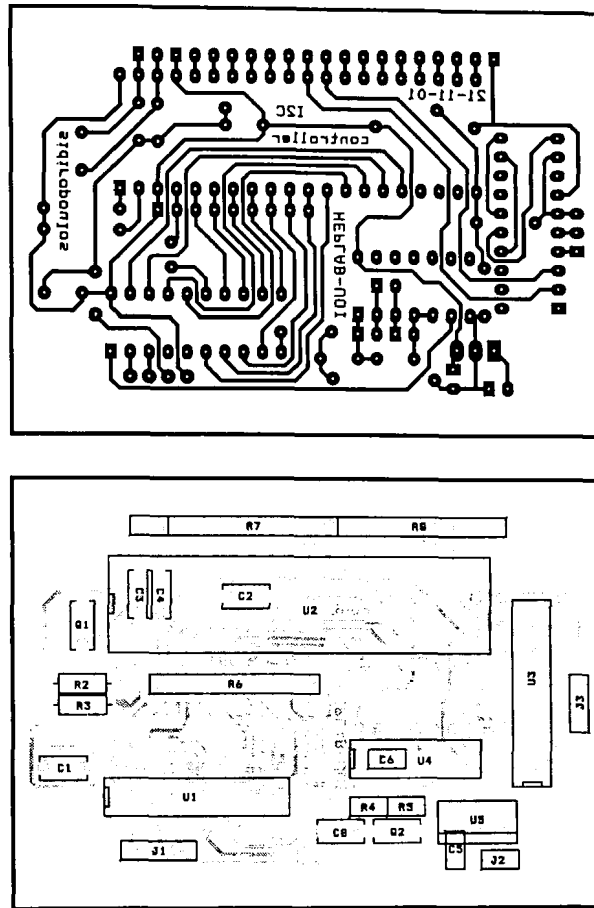
Πίνακας Β-1. Ο κατάλογος των υλικών της κάρτας PCI.



High Energy Physics Lab - University of Ioannina		
Title	RS-232 I2C Controller	Rev 1
Designer	Sidiropoulos George	Item

Σχήμα Β-5. Το ηλεκτρονικό σχέδιο του RS-232 I²C Controller.





Σχήμα Β-6. Το τυπωμένο κύκλωμα του RS-232 I²C controller και το τοπογραφικό τοποθέτησης των ηλεκτρονικών εξαρτημάτων

Υλικό	Τιμή	Περιγραφή
C1 - C2	100nf	Πυκνωτής πολυεστ.
C3 - C4	22pF	Πυκνωτής κεραμ.
C5 - C7	100nf	Πυκνωτής πολυεστ.
C8	100pF	Πυκνωτής κεραμ.
J1	Σύνδεσμος	4 pin-array
J2	Σύνδεσμος	2 pin-array
J3	Σύνδεσμος	3 pin-array
Q1	4MHz	Κρύσταλλος
Q2	8MHz	Κρύσταλλος
R1	10M	Αντιστάτης 1/4 W
R2 - R3	4,7 KΩ	Αντιστάτης 1/4 W
R4 - R5	360 Ω	Αντιστάτης 1/4 W
R6 - R8	4,7 KΩ	Αντιστάτης network
R9 - R10	4,7 KΩ	Αντιστάτης 1/4 W
U1	PCF8584	PCI I2C Controller
U2	68HC0705C8A	Μικροελεγκτής
U3	MAX233	Διεπαφή RS-232
U4	74LS04	Αντιστροφέας
U5	7805	Σταθεροποιητής τάσης

Πίνακας Β-2. Ο κατάλογος των υλικών της κάρτας PCI.

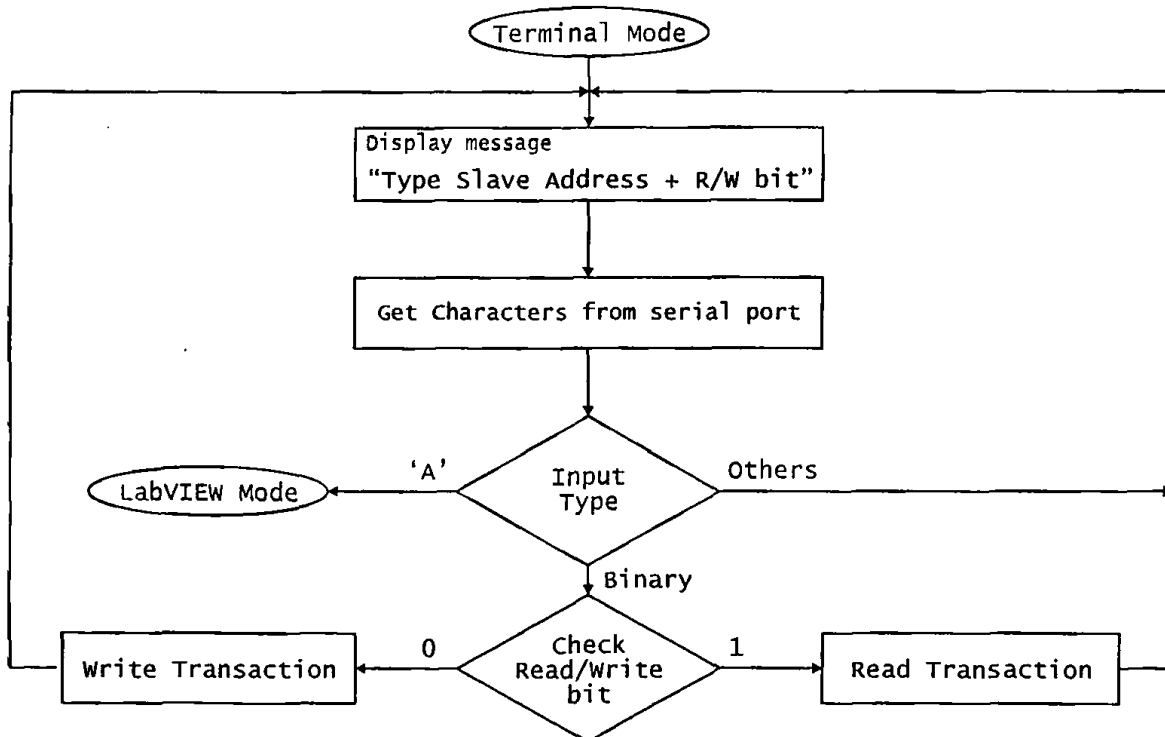


ΠΑΡΑΡΤΗΜΑ Γ

Προγραμματισμός του μικροελεγκτή σε γλώσσα Assembly

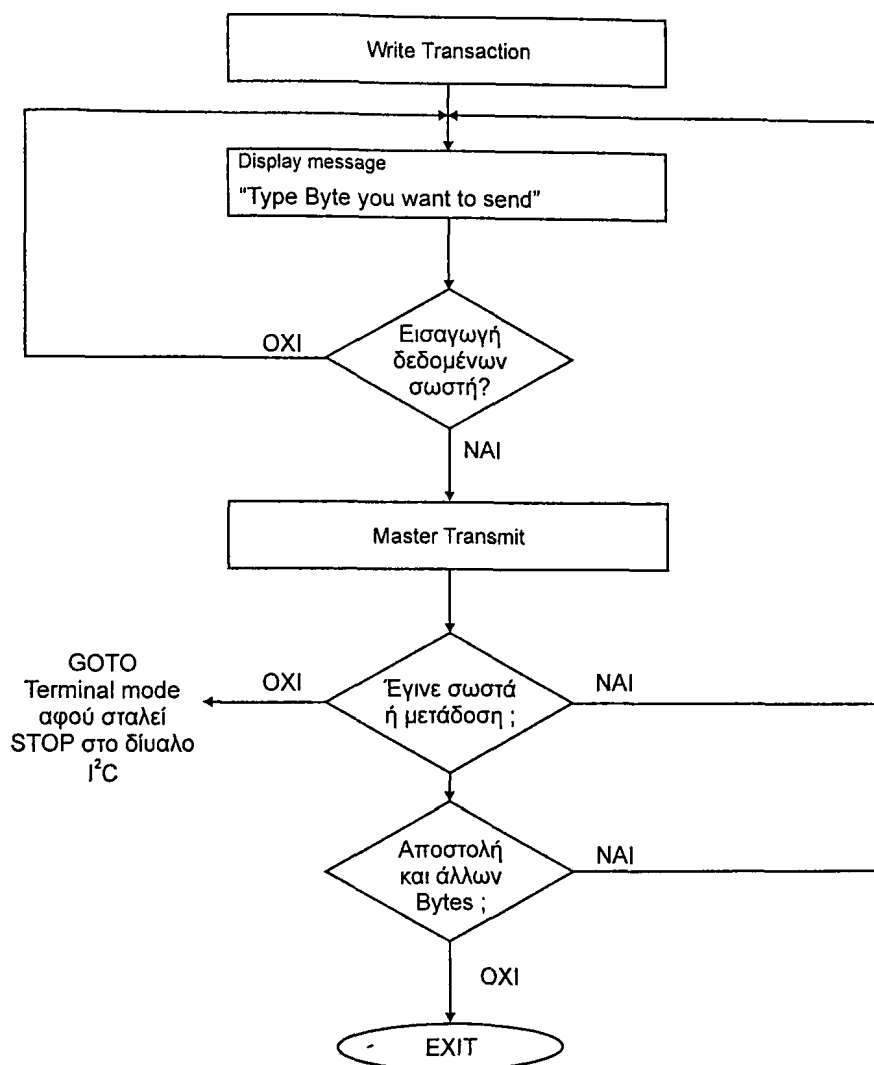
Στο παράρτημα αυτό παρατίθεται ο κώδικας με τον οποίο προγραμματίστηκε ο μικροελεγκτής 68HC706C8A. Για την κατανόηση των υποπρογραμμάτων παραθέτονται στη συνέχεια γενικά διαγράμματα ροής. Επιπλέον για την κατανόηση του κώδικα μπορούν να χρησιμοποιηθούν τα αναλυτικότερα διαγράμματα ροής των σχημάτων 4-6, 4-7 και 4-8.

Terminal mode. Κατά την εκκίνηση του ο μικροελεγκτής βρίσκεται σε λειτουργία Terminal mode και αποστέλλει μέσω της RS-232 το μήνυμα "Type Control Byte [7bit Slave Address + R/W in binary]" που είδαμε στην εικόνα 5-3. Στη συνέχεια μόλις ο χρήστης εισάγει το control byte, ο μC ελέγχει τους χαρακτήρες που έλαβε μέσω της σειριακής θύρας. Στην περίπτωση που ο πρώτος χαρακτήρας είναι 'A' (πρόκειται για το πρωτόκολλο που περιγράφηκε στη παράγραφο 5.3.2.1) ο μC μεταβαίνει σε λειτουργία LabVIEW. Διαφορετικά ελέγχει τα δεδομένα του control byte και ανάλογα με το τελευταίο bit R/W εκτελεί το αντίστοιχο υποπρόγραμμα, Read Transaction η Write transaction, όπως φαίνεται στο σχήμα Γ-1. Τα διαγράμματα ροής των υποπρογραμμάτων Write transaction και Read Transaction, δίνονται στα σχήματα Γ-2 και Γ-3 αντίστοιχα.

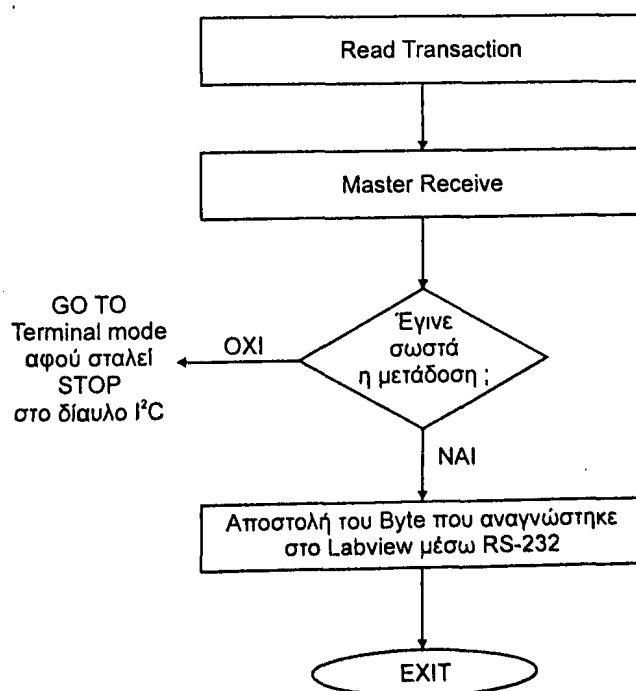


Σχήμα Γ-1. Το διάγραμμα ροής της λειτουργίας "Terminal Mode".

LabVIEW mode. Κατά τη λειτουργία LabVIEW ο μικροελεγκτής παρακολουθεί τα δεδομένα που στέλνει το πρόγραμμα LabVIEW μέσω της θύρας RS-232 και εκτελεί τις αντίστοιχες υπορουτίνες που φαίνονται στο σχήμα Γ-4 σύμφωνα με το ιδιαίτερο πρωτόκολλο που περιγράφηκε στην παράγραφο 5.3.2.1

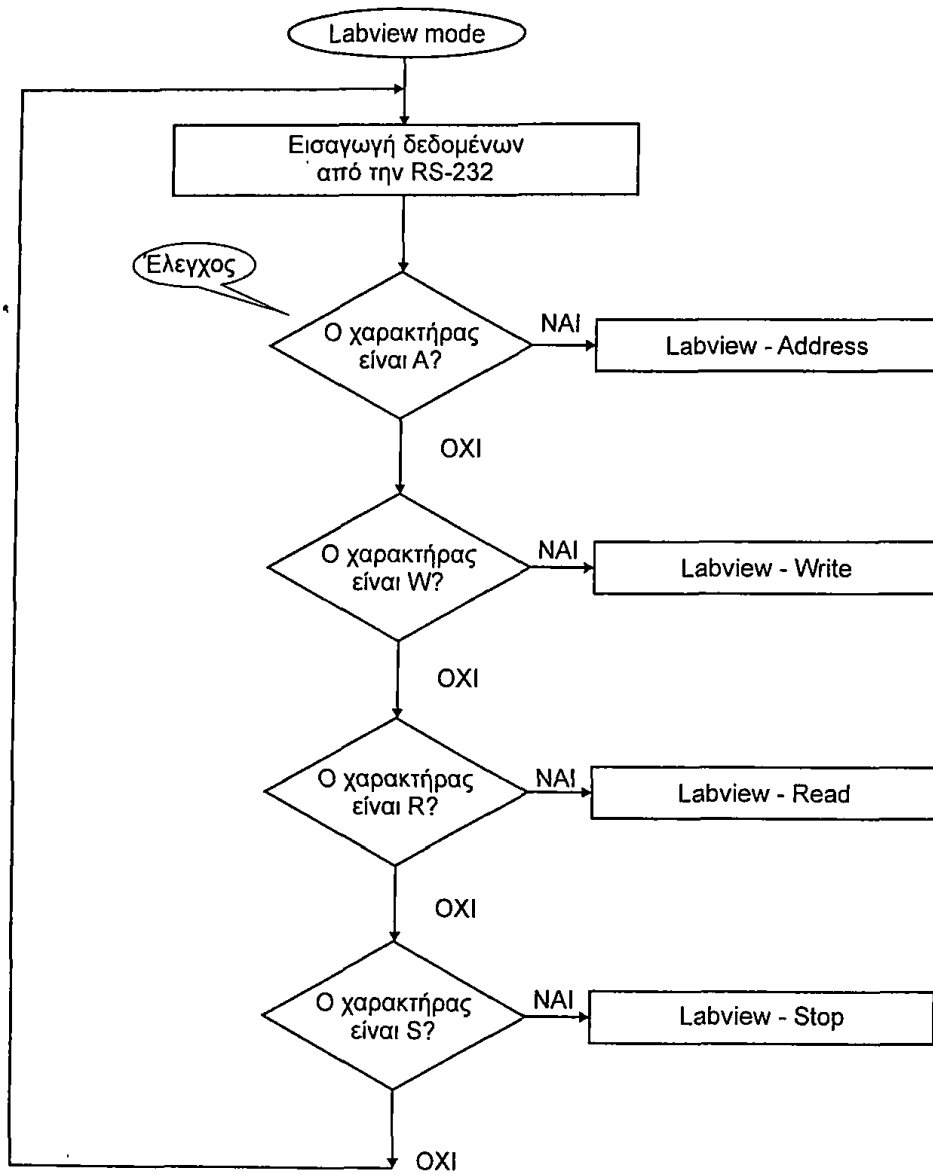


Σχήμα Γ-2. Το διάγραμμα ροής του υποπρογράμματος "Write Transaction".



Σχήμα Γ-3. Το διάγραμμα ροής του υποπρογράμματος "Read Transaction".





Σχήμα Γ-4. Το διάγραμμα ροής της λειτουργίας "LabVIEW Mode".

Ο Κώδικας σε γλώσσα Assembly

```

;*****
;*****RS-232 i2c Controller*****
;mc name : MC68HC705
;
;revision :3
;date :1/12/2001
;
;author : George Sidiropoulos
;filename :i2c.asm
;included files : subs.asm & sci.asm

;
#include "subs.asm"
#include "sci.asm" ;debugging

Mode_Register EQU $80
Address_Register EQU $81
Conversion_Reg EQU $82
First_Character EQU $83
Second_Character EQU $84

RW EQU 0

Start ORG $0160
LDA #$FF
STA PortA.Direction ;PortA (Data_Port)--> Output
STA PortB.Direction ;PortB --> Output

PowerUpDelay JSR Delay100ms
JSR Initialization ;Initialization Of PCF8584
JSR SCI_Initialize ; debugging

Select_Mode JSR GETdata
CMP #'A'
BEQ LabView
CMP #' '
BEQ Terminal_Mode
JMP Select_Mode

Terminal_Mode CLRX
SendMessage1 LDA Message1,X
JSR SENDData
INCX
CPX #!65
BNE SENDMessage1 ;Show welcome message

;*****
; "Type Control byte [7bit slave address + R/W] in binary"
;*****

Type_Address CLR i2cStatus
CLR

SendMessage2 LDA Message2,X
JSR SENDData
INCX
CPX #!58
BNE SENDMessage2

```



```

;***** Read Control Byte *****
Read_Address  CLR    Address_Register
               CLRX
ReadBit        JSR    GETdata
               CMP    #'A'
               BEQ    LabView
               CMP    #'0'
               BEQ    zero
               CMP    #'1'
               BEQ    one
               JMP    Type_Address      ;Error2
One            INC    Address_Register
Zero          INCX
               CPX    #!8
               BEQ    Send_Address
               LSL    Address_Register
               JMP    ReadBit

Send_Address  CLR    i2cStatus
               LDA    Address_Register
               STA    Slave_Address
               JSR    Addressing
               BRCLR  EF,i2cStatus,Send_or_Receive
               JMP    Stop_Condition
LabView       JMP    LabView_Address

;*****
; "Press S to send a byte or R to receive a byte"
;*****

Send_or_Receive BRCLR RW,Slave_address,Send_Mode
               CLRX
SendMessage3   LDA    Message3,X
               JSR    SENDData
               INCX
               CPX    #!49
               BNE    SENDMessage3

;***** Read_S_or_R *****
Read_S_or_R   JSR    GETdata
               CMP    #'S'
               BEQ    Send_Mode
               CMP    #'s'
               BEQ    Send_Mode
               CMP    #'R'
               BEQ    Receive_Mode
               CMP    #'r'
               BEQ    Receive_Mode
               JMP    Send_or_Receive

;***** Receive_Mode *****

Receive_Mode  JSR    Master_Receive
               BRSET  EF,i2cStatus,Stop_Condition
               LDA    Slave_Data
               STA    Conversion_Reg
               JSR    HEX_to_ASCII      ;Convert read Read to ASCII

               LDA    #$0D              ;SEND Enter

```



```

        JSR    SENDData
        LDA    #$0A
        JSR    SENDData
        LDA    First_Character      ;Send The read byte
        JSR    SENDData
        LDA    Second_Character
        JSR    SENDData
        JMP    Type_address        ;One_read only
;*****
;
;      "Type byte to send [hex]"
;*****
Send_Mode    CLRX
SendMessage4 LDA    Message4,X
             JSR    SENDData
             INCX
             CPX    #!27
             BNE    SENDMessage4

;***** get number *****

        JSR    Getdata
        STA    First_Character
        CMP    #'0'
        BLO    Send_Mode
        CMP    #'f'
        BHI    Send_Mode
        JSR    Getdata
        STA    Second_Character
        CMP    #'0'
        BLO    Send_Mode
        CMP    #'f'
        BHI    Send_Mode

        JSR    ASCII_to_byte      ;convert 2 characters
        STA    Slave_Data
        JSR    Master_Transmit    ;to one byte
        BRSET EF,i2cStatus,Stop_Condition

;*****
;
;      "Continue? [Y/N]"
;*****

SendMessage6 CLRX
             LDA    Message6,X      ; No Ack
             JSR    SENDData
             INCX
             CPX    #!19
             BNE    SENDMessage6

Read_Y_or_N  JSR    GETdata
             CMP    #'Y'
             BEQ    yes
             CMP    #'y'
             BEQ    yes
             CMP    #'N'
             BEQ    Stop_Condition
             CMP    #'n'
             BEQ    Stop_Condition
             JMP    Continue

Yes          BRCLR  RW,Slave_address,Write_mode

```



```
Read_mode    JMP    Send_or_Receive
Write_mode   JMP    Send_mode
```

***** Stop Condition *****

```
Stop_Condition  JSR    Stop
                BRCLR  EF,i2cStatus,complete
```

```
No_Ack        CLRX
SendMessage5   LDA    Message5,X          ; No Ack
                JSR    SENDData
                INCX
                CPX    #!30
                BNE    SENDMessage5
```

```
complete      JMP    Type_Address
```

***** Messages *****

```
Message1      FCB    $0D,$0A,'HEPLAB - University Of Ioannina',$0D,$0A ;31
chars + 4
                FCB    '      i2c Master Controller',$0D,$0A,$0D,$0A      ;26
chars + 4
```

```
Message2      FCB    $0D,$0A,'Type Control byte [7bit slave address + R/W] in
binary',$0D,$0A ;54 chars + 4
```

```
Message3      FCB    $0D,$0A,'Press S to send a byte or R to receive a
byte',$0D,$0A ;45 chars + 4
```

```
Message4      FCB    $0D,$0A,'Type byte to send [hex]',$0D,$0A
;23 chars + 4
```

```
Message5      FCB    $0D,$0A,'Error - No acknowledgement',$0D,$0A
;26 chars + 4
```

```
Message6      FCB    $0D,$0A,'Continue? [Y/N] ', $0D,$0A
;15 chars + 4
```

```
;SENDData    STA    $!1          ;Dummy
;            RTS              ;DUMMY
;GetData      RTS              ;DUMMY
```

***** ASCII Conversions *****

```
HEX_to_ASCII  LSRA
                LSRA
                LSRA
                LSRA
                CMP    #$0A
                BLO    Nine_to_Zero1
                ADD    #$07
Nine_to_Zero1  ADD    #$30
                STA    First_Character
                LDA    Conversion_Reg
                AND    #%00001111
                CMP    #$0A
                BLO    Nine_to_Zero2
                ADD    #$07
Nine_to_Zero2  ADD    #$30
                STA    Second_Character
                RTS
```



```

ASCII_to_HEX    CMP    #'F'
                BLS    Caps_or_number1
                SUB    #$20
Caps_or_number1 CMP    #'A'
                BLO    Nine_to_zero3
                SUB    #$07
Nine_to_zero3   SUB    #$30
                RTS

ASCII_to_byte   LDA    second_character
                JSR    ASCII_to_HEX
                STA    Conversion_Reg
                LDA    first_character
                JSR    ASCII_to_HEX
                LSLA
                LSLA
                LSLA
                LSLA
                ADD    Conversion_Reg
                RTS

```

;***** LabView Mode *****

```

                ORG    $400

```

;***** Get Command *****

```

LabView_Mode    CLR    i2cStatus

Polling         BCLR   EF,i2cStatus
                JSR    Getdata
                CMP    #'A'
                BEQ    LabView_Address
                CMP    #'W'
                BEQ    LabView_Write
                CMP    #'R'
                BEQ    LabView_Read
                CMP    #'S'
                BEQ    LabView_Stop
                JMP    Polling

```

;***** LabView_Address *****

```

LabView_Address CLR    i2cStatus
                JSR    Getdata
                STA    First_Character
                JSR    Getdata
                STA    Second_Character
                JSR    ASCII_to_Byte
                STA    Slave_Address
                JSR    Stop
                JSR    Addressing
                BRSET  EF,i2cStatus,LabView_Stop

                LDA    #'C'
                JSR    Sendata

                JMP    Polling

```

;Success Message



```

;***** LabView_Write *****
LabView_Write   JSR    Getdata
                 STA    First_Character
                 JSR    Getdata
                 STA    Second_Character
                 JSR    ASCII_to_Byte
                 STA    Slave_Data
                 JSR    Master_Transmit
                 BRSET  EF,i2cStatus,LabView_Stop

                 LDA    #'C'
                 JSR    Senddata

                 JMP    Polling

;***** LabView_Read *****
LabView_Read    BRCLR  RV,i2cStatus,Simple_Receive
                 JSR    Stop
                 JSR    Addressing
Simple_Receive  JSR    Master_Receive
                 BRSET  EF,i2cStatus,LabView_Stop

                 LDA    #'C'                                ;Success Message
                 JSR    Senddata

                 LDA    Slave_Data
                 STA    Conversion_Reg
                 JSR    HEX_to_ASCII                        ;Convert read Read to
ASCII
                 LDA    First_Character                    ;Send The read byte
                 JSR    SENData
                 LDA    Second_Character
                 JSR    SENData
                 JMP    Polling

;***** LabView_Stop *****
LabView_Stop    JSR    Stop
                 BRCLR  EF,i2cStatus,No_error
                 LDA    #'E'
                 JSR    SENData                            ;Error Message
No_Error        JMP    LabView_Mode

;*****
                 ORG    $1ffe
                 FDB    $0160

;*****

```



```

;*****
;*****RS-232 i2c Controller*****
;mc name : MC68HC705
;
;revision :3
;date :1/12/2001
;
;author : George Sidiropoulos
;filename : subs.asm
;included files : i2c.asm & sci.asm

```

```

; ***** BYTE DEFINITION *****

```

```

PortA.Data EQU $00
PortB.Data EQU $01
Data_Port EQU $00
Control_Port EQU $01

```

```

PortA.Direction EQU $04
PortB.Direction EQU $05
Data_P.Direction EQU $04
Control_P.Direction EQU $05

```

```

;Byte Definition

```

```

DataSend EQU $50
DataRead EQU $51
i2cStatus EQU $52

```

```

Slave_Address EQU $60
Slave_Data EQU $61

```

```

NumberOfBytes EQU $70
ByteRead EQU $71

```

```

tempA EQU $85
tempX EQU $86

```

```

;***** Bit Definition *****

```

```

LED EQU 7 ;Pin7 PortB
RESET EQU 4 ;Pin4 PortB
WR EQU 3 ;Pin3 PortB
CS EQU 2 ;Pin2 PortB
RD EQU 1 ;Pin1 PortB
A0 EQU 0 ;Pin0 PortB

```

```

BB EQU 0 ;Bit0 Status Register
LRB EQU 3 ;Bit3 Status Register
PIN EQU 7 ;Bit7 Status Register

```

```

EF EQU 0 ;Error Flag (bit0 i2c Status)
DR EQU 1 ;Dummy Read (bit1 i2c Status)
RV EQU 2 ;Receive Flag (bit2 i2c Status)

```

```

EP EQU 3 ;Error PIN
EB EQU 4 ;Error Bus Is Busy

```

```

ORG $600

```



***** INITIALIZATION *****

```

Initialization LDA    #%11001111          ;LED->0,Reset->Low,WR ->High,CS-
                STA    Control_Port        ;>High,RD-->High,A0->High
                JSR    Delay100ms
                BSET   RESET,Control_Port  ;deassert RESET
                BSET   A0,Control_Port     ;access S1
                LDA    #$80                ;ES0<-0 [I2C disable] ES1<-0,ES2<-
                ;0==>next W to S0'
                JSR    WriteCycle
                BCLR   A0,Control_Port     ;write to S0'
                LDA    #$55                ;effective own address AAh
                JSR    WriteCycle
                BSET   A0,Control_Port     ;access S1
                LDA    #$A0                ;ES1<-1,ES2<-0==>next W to S2
                JSR    WriteCycle
                BCLR   A0,Control_Port     ;write to S2
                LDA    #%00011000;         ;System Clock 8MHz, SCL=90KHz
                JSR    WriteCycle
                BSET   A0,Control_Port     ;access S1
                LDA    #$C1                ;ES0<-1 [I2C enabled but idle mode]
                ;ACK<-1 [normal]
                JSR    WriteCycle

                JSR    Delay100ms
                JSR    Delay100ms
    
```

RTS

***** Addressing Subroutine *****

```

Addressing      BSET   A0,Control_Port     ;Access S1
BusBusy        JSR    ReadCycle            ;Read Status Register
                BRCLR  BB,DataRead,BusBusy ;Loop Until BB=1(deasserted)
                BCLR   A0,Control_Port     ;access S0
                LDA    Slave_Address       ;Write Slave Address (7bit
+R/W=0) to S0  JSR    WriteCycle
                BSET   A0,Control_Port     ;access S1
                LDA    #$C5                ;Send Start Condition+Slave Address
                JSR    WriteCycle
                JSR    Ack
Exit_address    RTS
    
```

***** Master Transmit *****

```

Master_Transmit BCLR   A0,Control_Port     ;Access S0
                LDA    Slave_Data
                JSR    WriteCycle
                JSR    Ack
                RTS
    
```

***** Stop Condition *****

```

Stop           BSET   A0,Control_Port
                LDA    #$C3                ;Stop Condition
                JSR    WriteCycle
                RTS
    
```



```

;***** Master Receive*****
Master_Receive  BSET    RV,i2cStatus

Negative_ack    BSET    A0,Control_Port    ;access S1
                LDA     #$40                ;ACK bit --> 0
                JSR     WriteCycle
                BCLR   A0,Control_Port
                JSR     ReadCycle           ;Receive the final data buffer
                STA     Slave_Data

Read_PIN        BSET    A0,Control_Port
                JSR     ReadCycle
                BRSET  PIN,DataRead,Read_PIN ;Loop Until PIN=0
                BSET  A0,Control_Port
                LDA     #$C3                ;Stop Condition
                JSR     WriteCycle
                BCLR  A0,Control_Port      ;access S0
                JSR     ReadCycle           ;Transfer final data byte from
data
                STA     Slave_Data         ;buffer to accumulator
                RTS

***** Write Cycle *****
WriteCycle      STA     DataSend
                LDA     #$FF                ;
                STA     Data_P.Direction    ;Data Port --> Output
                BSET  RD,Control_Port      ;Disable Read Mode
                BCLR  CS,Control_Port      ;Assert CS
                LDA     DataSend           ;Read From Register DataSend
                STA     Data_Port          ;Data To Transfer Ready
                BCLR  WR,Control_Port      ;WR Assert
                BSET  WR,Control_Port      ;WR Deassert
                BSET  CS,Control_Port      ;Deassert CS
                RTS

;***** Read Cycle *****
ReadCycle       CLR     Data_P.Direction    ;Data_Port --> Input
                BSET  WR,Control_Port      ;Disable Write Mode
                BCLR  CS,Control_Port      ;Assert CS
                BCLR  RD,Control_Port      ;Assert RD
                LDA     Data_Port          ;Read Data Port
                STA     DataRead           ;Save to DataRead Register
                BSET  RD,Control_Port      ;Deassert RD
                BSET  CS,Control_Port      ;Deassert CS
                RTS

;***** Slave Acknowledgement *****
Ack             BSET    A0,Control_Port
                JSR     ReadCycle
                BRSET  PIN,DataRead,Ack    ;Loop Until PIN=0
                BRSET  LRB,DataRead,ErrorFlag ;Slave Acknowledgement
                BCLR  EF,i2cStatus        ;clear error flag
                RTS
ErrorFlag       BSET    EF,i2cStatus
                RTS

```



;***** Delay100ms*****

```

Delay100ms
DebounceDelay  STA    TempA          ;
                STX    TempX          ;
                LDX    #!100          ;
MiddleLoop     LDA    #!166          ;
InnerLoop      NOP                    ;
                NOP                    ;
                NOP                    ;
                DECA                   ;
                BNE    InnerLoop       ;
                DECX                   ;
                BNE    MiddleLoop      ;
                LDA    TempA           ;
                LDX    TempX           ;
                RTS
                ORG    $1ff0
                FDB    $0000
    
```

;*****

;*****
;*****RS-232 i2c Controller*****

```

;mc name   : MC68HC705
;
;revision  : 3
;date      : 1/12/2001
;
;author    : George Sidiropoulos
;filename  : sci.asm
;included  files : subs.asm & i2c.asm
    
```

```

BAUD_RATE    EQU    $0D
SCCR1        EQU    $0E
SCCR2        EQU    $0F
SCSR         EQU    $10
SCDAT        EQU    $11

TDRE         EQU    7
RDRF         EQU    5

                ORG    $500

SCI_Initialize LDA    #%00110000
                STA    BAUD_RATE      ;9600bps
                LDA    #$00
                STA    SCCR1          ;set up SCCR1
                LDA    #%00001100
                STA    SCCR2          ;bidirectional Communication
                RTS

GETDATA       BRCLR  RDRF,SCSR,GETDATA
                LDA    SCDAT
                RTS

SENDATA       BRCLR  TDRE,SCSR,SENDATA
                STA    SCDAT
                RTS
    
```



ΠΑΡΑΡΤΗΜΑ Δ

Τα σήματα του διαύλου PCI

Στον πίνακα Δ-1 φαίνονται όλα τα σήματα του διαύλου PCI εκ των οποίων τα κυριότερα περιγράφονται στο παρόν παράρτημα.

Ακροδέκτης	Σύνδεσμος 5 V		Σύνδεσμος 3.3 V		Σχόλια
	Side B	Side A	Side B	Side A	
1	-12V	TRST#	-12V	TRST#	Αρχή του σύνδεσμου 32-bit
2	TCK	+12V	TCK	+12V	
3	Ground	TMS	Ground	TMS	
4	TDO	TDI	TDO	TDI	
5	+5V	+5V	+5V	+5V	
6	+5V	INTA#	+5V	INTA#	
7	INTB#	INTC#	INTB#	INTC#	
8	INTD#	+5V	INTD#	+5V	
9	PRSNT1#	Reserved	PRSNT1#	Reserved	
10	Reserved	+5V (I/O)	Reserved	+3.3V (I/O)	
11	PRSNT2#	Reserved	PRSNT2#	Reserved	
12	Ground	Ground	Εγκοπή σύνδεσμου		Διαχωριστική Εγκοπή 3.3 volt
13	Ground	Ground	Εγκοπή σύνδεσμου		
14	Reserved	3.3Vaux	Reserved	3.3Vaux	Διαχωριστική Εγκοπή 3.3 volt
15	Ground	RST#	Ground	RST#	
16	CLK	+5V (I/O)	CLK	+3.3V (I/O)	
17	Ground	GNT#	Ground	GNT#	
18	REQ#	Ground	REQ#	Ground	
19	+5V (I/O)	PME#	+3.3V (I/O)	PME#	
20	AD[31]	AD[30]	AD[31]	AD[30]	
21	AD[29]	+3.3V	AD[29]	+3.3V	
22	Ground	AD[28]	Ground	AD[28]	
23	AD[27]	AD[26]	AD[27]	AD[26]	
24	AD[25]	Ground	AD[25]	Ground	
25	+3.3V	AD[24]	+3.3V	AD[24]	
26	C/BE[3]#	IDSEL	C/BE[3]#	IDSEL	
27	AD[23]	+3.3V	AD[23]	+3.3V	
28	Ground	AD[22]	Ground	AD[22]	
29	AD[21]	AD[20]	AD[21]	AD[20]	
30	AD[19]	Ground	AD[19]	Ground	
31	+3.3V	AD[18]	+3.3V	AD[18]	
32	AD[17]	AD[16]	AD[17]	AD[16]	
33	C/BE[2]#	+3.3V	C/BE[2]#	+3.3V	
34	Ground	FRAME#	Ground	FRAME#	
35	IRDY#	Ground	IRDY#	Ground	
36	+3.3V	TRDY#	+3.3V	TRDY#	
37	DEVSEL#	Ground	DEVSEL#	Ground	
38	Ground	STOP#	Ground	STOP#	
39	LOCK#	+3.3V	LOCK#	+3.3V	
40	PERR#	Reserved*	PERR#	Reserved*	
41	+3.3V	Reserved*	+3.3V	Reserved*	
42	SERR#	Ground	SERR#	Ground	

Πίνακας Δ-1. Οι ακροδέκτες του συνδέσμου PCI

Ακροδέκτης	Σύνδεσμος 5 V		Σύνδεσμος 3.3 V		Σχόλια
	Side B	Side A	Side B	Side A	
43	+3.3V	PAR	+3.3V	PAR	
44	C/BE[1]#	AD[15]	C/BE[1]#	AD[15]	
45	AD[14]	+3.3V	AD[14]	+3.3V	
46	Ground	AD[13]	Ground	AD[13]	
47	AD[12]	AD[11]	AD[12]	AD[11]	
48	AD[10]	Ground	AD[10]	Ground	
49	Ground	AD[09]	M66EN	AD[09]	66 MHz / gnd
50	Εγκοπή σύνδεσμου		Ground	Ground	Διαχωριστική Εγκοπή 5 volt
51	Εγκοπή σύνδεσμου		Ground	Ground	Διαχωριστική Εγκοπή 5 volt
52	AD[08]	C/BE[0]#	AD[08]	C/BE[0]#	
53	AD[07]	+3.3V	AD[07]	+3.3V	
54	+3.3V	AD[06]	+3.3V	AD[06]	
55	AD[05]	AD[04]	AD[05]	AD[04]	
56	AD[03]	Ground	AD[03]	Ground	
57	Ground	AD[02]	Ground	AD[02]	
58	AD[01]	AD[00]	AD[01]	AD[00]	
59	+5V (I/O)	+5V (I/O)	+3.3V (I/O)	+3.3V (I/O)	
60	ACK64#	REQ64#	ACK64#	REQ64#	
61	+5V	+5V	+5V	+5V	
62	+5V	+5V	+5V	+5V	Τέλος του σύνδεσμου 32-bit
	Διάκενο		Διάκενο		64-bit spacer
	Διάκενο		Διάκενο		64-bit spacer
63	Reserved	Ground	Reserved	Ground	Αρχή του σύνδεσμου 64-bit
64	Ground	C/BE[7]#	Ground	C/BE[7]#	
65	C/BE[6]#	C/BE[5]#	C/BE[6]#	C/BE[5]#	
66	C/BE[4]#	+5V (I/O)	C/BE[4]#	+3.3V (I/O)	
67	Ground	PAR64	Ground	PAR64	
68	AD[63]	AD[62]	AD[63]	AD[62]	
69	AD[61]	Ground	AD[61]	Ground	
70	+5V (I/O)	AD[60]	+3.3V (I/O)	AD[60]	
71	AD[59]	AD[58]	AD[59]	AD[58]	
72	AD[57]	Ground	AD[57]	Ground	
73	Ground	AD[56]	Ground	AD[56]	
74	AD[55]	AD[54]	AD[55]	AD[54]	
75	AD[53]	+5V (I/O)	AD[53]	+3.3V (I/O)	
76	Ground	AD[52]	Ground	AD[52]	
77	AD[51]	AD[50]	AD[51]	AD[50]	
78	AD[49]	Ground	AD[49]	Ground	
79	+5V (I/O)	AD[48]	+3.3V (I/O)	AD[48]	
80	AD[47]	AD[46]	AD[47]	AD[46]	
81	AD[45]	Ground	AD[45]	Ground	
82	Ground	AD[44]	Ground	AD[44]	

Πίνακας Δ-1 (συνέχεια). Οι ακροδέκτες του συνδέσμου PCI



Ακροδέκτης	Σύνδεσμος 5 V		Σύνδεσμος 3.3 V		Σχόλια
	Side B	Side A	Side B	Side A	
83	AD[43]	AD[42]	AD[43]	AD[42]	
84	AD[41]	+5V (I/O)	AD[41]	+3.3V (I/O)	
85	Ground	AD[40]	Ground	AD[40]	
86	AD[39]	AD[38]	AD[39]	AD[38]	
87	AD[37]	Ground	AD[37]	Ground	
88	+5V (I/O)	AD[36]	+3.3V (I/O)	AD[36]	
89	AD[35]	AD[34]	AD[35]	AD[34]	
90	AD[33]	Ground	AD[33]	Ground	
91	Ground	AD[32]	Ground	AD[32]	
92	Reserved	Reserved	Reserved	Reserved	
93	Reserved	Ground	Reserved	Ground	
94	Ground	Reserved	Ground	Reserved	Τέλος του σύνδεσμου 64-bit

Πίνακας Δ-1 (συνέχεια). Οι ακροδέκτες του συνδέσμου PCI

CLK :

Το σήμα CLK παρέχει τη χρονική αναφορά (σήμα συγχρονισμού) για όλες τις μεταφορές δεδομένων στο δίαυλο PCI. Όλα τα σήματα του PCI εκτός του reset και των interrupts δειγματοληπτούνται κατά τη μετάβαση από λογικό '0' σε λογικό '1' (rising edge) του σήματος CLK. Όλα τα χρονικά χαρακτηριστικά (timing specifications) ορίζονται με βάση τη μετάβαση από λογικό '0' σε λογικό '1' του σήματος CLK. Η μέγιστη συχνότητα του σήματος CLK στα περισσότερα συστήματα PCI είναι 66MHz. Ως ελάχιστη συχνότητα του σήματος CLK ορίζονται τα 0 Hz, επιτρέποντας ουσιαστικά την απενεργοποίηση (suspend) του σήματος CLK για λόγους εξοικονόμησης ενέργειας.

RST# :

Το σήμα RST# οδηγείται σε λογικό '0' για να επαναφέρει στην αρχική κατάσταση μια συσκευή PCI, αρχικοποιώντας τους καταχωρητές ρυθμίσεων, τις μηχανές καταστάσεων (state machines) και τα σήματα εξόδου. Το σήμα RST# μπορεί να μεταβάλλεται ασύγχρονα σε σχέση με το CLK. Παραμένει ενεργό για τουλάχιστο 100μsec από τη στιγμή που σταθεροποιείται το σήμα CLK.

AD[31:0] :

Οι δίαυλοι διευθυνσιοδότησης (address) και δεδομένων (data) πολυπλέκονται σε αυτά τα σήματα (pins). Ο δίαυλος AD[31:0] μεταφέρει κατά τη διάρκεια της φάσης διευθυνσιοδότησης (address phase) τη διεύθυνση εύρους 32-bit ενώ κατά τη διάρκεια της φάσης μεταφοράς δεδομένων (data phase) τα δεδομένα εύρους 32-bit.

C/BE[3:0]# :

Σε αυτά τα σήματα πολυπλέκονται τα σήματα εντολών (bus command) και επιλογής byte (byte enable). Κατά τη διάρκεια της φάσης διευθυνσιοδότησης τα σήματα αυτά μεταφέρουν την εντολή που δηλώνει τον τύπο της μεταφοράς που θα λάβει χώρα. Στον πίνακα Δ-2 απεικονίζονται οι πιθανές τιμές που μπορεί να πάρει ο δίαυλος και ο τύπος της μεταφοράς στον οποίο αντιστοιχούν. Κατά τη διάρκεια της μεταφοράς

δεδομένων τα σήματα αυτά δηλώνουν ποια από τα byte στο διάυλο AD[31:0] είναι έγκυρα. Το σήμα C/BE[3]# αντιστοιχεί στο σημαντικότερο byte (AD[31:24]) ενώ το σήμα C/BE[0]# αντιστοιχεί στο λιγότερο σημαντικό byte (AD[7:0]). Τα σήματα C/BE[3:0]# οδηγούνται από τον Initiator καθ' όλη τη διάρκεια της μεταφοράς δεδομένων.

C/BE[3:0]#	Τύποι εντολών
0000	Επιβεβαίωση Διακοπής
0001	Ειδικός κύκλος
0010	Εγγραφή περιοχής I/O
0011	Ανάγνωση περιοχής I/O
0100	Δεσμευμένο
0101	Δεσμευμένο
0110	Ανάγνωση περιοχής μνήμης
0111	Εγγραφή περιοχής μνήμης
1000	Δεσμευμένο
1001	Δεσμευμένο
1010	Ανάγνωση περιοχής ρυθμίσεων
1011	Εγγραφή περιοχής ρυθμίσεων
1100	Πολλαπλή ανάγνωση μνήμης
1101	Διπλός κύκλος διευθυνσιοδότησης
1110	Memory Read Line
1111	Memory Write and Invalidate

Πίνακας Δ-2. Οι εντολές συναλλαγής στο διάυλο PCI.

PAR :

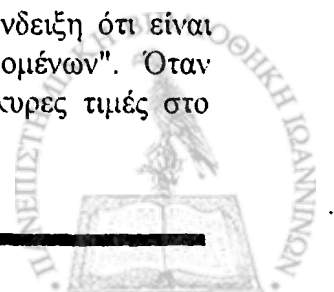
Το σήμα PAR αποτελεί την άρτια ισοτιμία των σημάτων AD[31:0] και C/BE[3:0]#. Κατά την άρτια ισοτιμία, ο συνολικός αριθμός των '1' στα σήματα AD[31:0], C/BE[3:0]# και PAR είναι άρτιος. Το σήμα PAR έχει τα ίδια χρονικά χαρακτηριστικά με τα σήματα AD[31:0] με τη μόνη διαφορά ότι καθυστερεί έναν κύκλο ρολογιού για να δοθούν τα χρονικά περιθώρια για τον υπολογισμό της ισοτιμίας.

FRAME# :

Το σήμα FRAME# οδηγείται σε λογικό '0' από τον "initiator" σηματοδοτώντας μια νέα μεταφορά δεδομένων μέσω του διαύλου. Η φάση διευθυνσιοδότησης ("address phase") στον πρώτο παλμό ρολογιού μετά από μετάφραση του σήματος FRAME# από λογικό '1' σε λογικό '0'. Αν ο "initiator" σκοπεύει να πραγματοποιήσει μια φάση μεταφοράς δεδομένων ("data phase"), οδηγεί το σήμα FRAME# ξανά σε λογικό '1', μετά από ένα κύκλο ρολογιού. Αν πρόκειται να εκτελεστούν πολλαπλές φάσεις μεταφοράς δεδομένων, ο "initiator" διατηρεί το σήμα FRAME# σε λογικό '0' κατά τη διάρκεια όλων των φάσεων, εκτός από την τελευταία.

IRDY# :

Το σήμα IRDY# οδηγείται σε λογικό '0' από τον "initiator" σαν ένδειξη ότι είναι έτοιμος να ολοκληρώσει την τρέχουσα "φάση μεταφοράς δεδομένων". Όταν πρόκειται για εγγραφή δεδομένων δηλώνει ότι έχουν οδηγηθεί έγκυρες τιμές στο



διάυλο AD[31:0] ενώ όταν πρόκειται για ανάγνωση δηλώνει ότι είναι έτοιμος να δεχθεί δεδομένα στο διάυλο AD[31:0].

TRDY# :

Το σήμα IRDY# οδηγείται σε λογικό '0' από τον "target" σαν ένδειξη ότι είναι έτοιμος να ολοκληρώσει την τρέχουσα "φάση μεταφοράς δεδομένων". Όταν πρόκειται για εγγραφή δεδομένων δηλώνει ότι είναι έτοιμος να δεχθεί δεδομένα στο διάυλο AD[31:0] ενώ όταν πρόκειται για ανάγνωση, έχουν οδηγηθεί έγκυρες τιμές στο διάυλο AD[31:0]

STOP# :

Το σήμα STOP# , όταν οδηγείται σε λογικό '0' από τον "target", δηλώνει αίτηση προς τον "initiator" για τερματισμό της τρέχουσας μεταφοράς δεδομένων.

IDSEL :

Το σήμα IDSEL χρησιμοποιείται ως "chip select" κατά τη διάρκεια προσπέλασης της περιοχής ρυθμίσεων ("configuration space"). Το σήμα IDSEL οδηγείται από το σύστημα PCI και υπάρχει ένα για κάθε υποδοχή PCI. Αυτό το χαρακτηριστικό δίνει τη δυνατότητα στο μηχανισμό ρύθμισης του PCI να προσπελάσει ξεχωριστά κάθε συσκευή PCI του συστήματος.

DEVSEL# :

Το σήμα DEVSEL οδηγείται σε λογικό '0' από τον "target" όταν αυτός αναγνωρίσει τη διεύθυνση του στο διάυλο PCI.

REQ# :

Το σήμα REQ# χρησιμοποιείται από τις "master" συσκευές PCI όταν αυτές ζητούν την κυριότητα του διαύλου PCI. Τα σήματα REQ# από κάθε συσκευή PCI λαμβάνονται από τον διαιτητή του διαύλου ("arbiter").

GNT# :

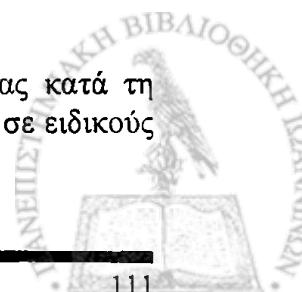
Το σήμα GNT# οδηγείται από τον διαιτητή του διαύλου και κατοχυρώνει την κυριότητα του διαύλου σε κάποια από τις συσκευές PCI. Το σήμα GNT# είναι μοναδικό για κάθε συσκευή PCI.

PERR# :

Το σήμα PERR# χρησιμοποιείται για την αναφορά σφάλματος ισοτιμίας κατά την ανταλλαγή δεδομένων. Οδηγείται σε λογικό '0' δύο κύκλους μετά τη φάση μεταφοράς δεδομένων με τη λανθασμένη ισοτιμία.

SERR# :

Το σήμα SERR# χρησιμοποιείται για την αναφορά σφάλματος ισοτιμίας κατά τη διευθυνσιοδότηση, σφάλματος ισοτιμίας κατά την ανταλλαγή δεδομένων σε ειδικούς κύκλους (Special cycle) ή κάποιου άλλου κρίσιμου σφάλματος.



INTA#, INTB#, INTC#, INTD# :

Τα σήματα διακοπής οδηγούνται σε λογικό '0' από τις συσκευές PCI για να επιστήσουν την προσοχή του λογισμικού οδήγησής τους. Είναι "level sensitive" και οδηγούνται σε λογικό '0' όπως τα σήματα "open drain signal". Από τη στιγμή που ενεργοποιηθούν, παραμένουν ενεργά μέχρι το λογισμικό οδήγησης να ικανοποιήσει την αίτησή τους.

PRSNT[1:2]# :

Τα σήματα αυτά χρησιμοποιούνται για δύο λόγους: Για να υποδείξουν ότι μια προσαρτώμενη κάρτα PCI είναι παρούσα σε μία υποδοχή και για να δηλώσουν τις απαιτήσεις σε ενέργεια της εν λόγω κάρτας. Πρόκειται για στατικά σήματα τα οποία είτε γειώνονται είτε μένουν ασύνδετα στην προσαρτώμενη κάρτα. Στον πίνακα Δ-3 απεικονίζεται η κωδικοποίηση αυτών των σημάτων.

PRSNT1#	PRSNT2#	Προσαρτώμενη κάρτα
Ασύνδετο	Ασύνδετο	Δεν υπάρχει κάρτα
Γειωμένο	Ασύνδετο	Κάρτα μέγιστης ισχύος 25W
Ασύνδετο	Γειωμένο	Κάρτα μέγιστης ισχύος 15W
Γειωμένο	Γειωμένο	Κάρτα μέγιστης ισχύος 7.5W

Πίνακας Δ-3. Οι διάφορες καταστάσεις των ακροδεκτών PRSNT[1:2]#.

M66EN :

Το σήμα αυτό παραμένει ασύνδετο σε κάρτες που υποστηρίζουν συχνότητα λειτουργίας 66MHz ενώ γειώνεται σε αυτές που λειτουργούν στα 33MHz. Στα συστήματα που υποστηρίζουν λειτουργία στα 66 MHz συνδέεται μια pull-up αντίσταση σε αυτό το σήμα για να ανιχνεύσει τη συχνότητα λειτουργίας της κάρτας. Αν το σήμα είναι σε λογικό '1', τότε η συχνότητα του CLK φτάνει τα 66MHz, ενώ στην αντίθετη περίπτωση η συχνότητα του CLK φτάνει τα 33MHz. Στα συστήματα που υποστηρίζουν λειτουργία στα 33 MHz το σήμα αυτό γειώνεται.

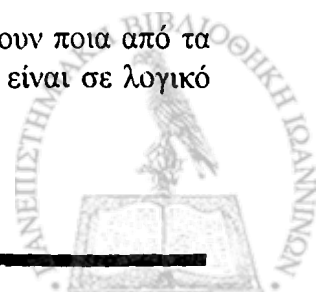
AD[63:32] :

Οι δίαυλοι διευθυνσιδότησης (address) και δεδομένων (data) πολυπλέκονται σε αυτά τα σήματα (pins) και παρέχουν 32 επιπλέον σήματα στα συστήματα που υποστηρίζουν εύρος 64-bit. Κατά τη διάρκεια των φάσεων μεταφοράς δεδομένων αυτά τα σήματα παρέχουν 32-bit δεδομένων επιπλέον, όταν τα σήματα REQ64# και Ack64# είναι σε λογικό '0'.

C/BE[7:4]# :

Σε αυτά τα σήματα πολυπλέκονται τα σήματα εντολών (bus command) και επιλογής byte (byte enable) και παρέχουν τέσσερα επιπλέον σήματα στα συστήματα που υποστηρίζουν εύρος 64-bit.

Κατά τη διάρκεια της μεταφοράς δεδομένων τα σήματα αυτά δηλώνουν ποια από τα byte στο δίαυλο AD[63:32], όταν τα σήματα REQ64# και ACK64# είναι σε λογικό '0'.



Κατά τη διάρκεια της φάσης διευθυνσιοδότησης, όταν εκτελείται διπλή διευθυνσιοδότηση και το σήμα REQ64#, αυτά τα σήματα μεταφέρουν την εντολή που δηλώνει τον τύπο μεταφοράς.

REQ64# :

- Το σήμα αυτό οδηγείται σε λογικό '0' από τον "initiator" για να δηλώσει μεταφορά δεδομένων 64-Bit. Ο χρονισμός του σήματος αυτού είναι ίδιος με αυτόν του σήματος FRAME#.

ACK64# :

Το σήμα αυτό οδηγείται σε λογικό '0' από κάποιον "target" ως ένδειξη ότι έχει αποκωδικοποιήσει την διεύθυνσή του και ότι είναι ικανό για μεταφορά δεδομένων εύρους 64-bit.

PAR64 :

Το σήμα PAR αποτελεί την άρτια ισοτιμία των σημάτων AD[63:32] και C/BE[7:4]#

JTAG (ΠΡΟΑΙΤΕΤΙΚΑ) :

Οι συσκευές PCI μπορούν προαιρετικά να υποστηρίζουν την διεπαφή IEEE 1149.1 JTAG / Boundary Scan. Το πρότυπο αυτό επιτρέπει την in-circuit δοκιμή και τον έλεγχο των εξαρτημάτων μιας κάρτας PCI. Σημειώνεται ότι αν δεν υλοποιείται το πρότυπο JTAG σε μια κάρτα τα pins TDO και TDI πρέπει να συνδεθούν μεταξύ τους.

TCK : *Test Clock*

TDI : *Test Data Input*

TDO : *Test Data Output*

TMS : *Test Mode Select*

TRST# : *Test Reset*

