

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΤΜΗΜΑ ΜΗΧ/ΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Εργασία στα Πλαίσια του μαθήματος:  
**ΔΙΑΣΥΝΔΕΣΗ ΜΙΚΡΟΪΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

Ακαδημαϊκό Έτος: 2009 – 2010

**Μπεχτσούδης Ανέστης**

***Επικοινωνία Spartan 3AN FPGA Board με PS/2 πληκτρολόγιο***



**Abstract:**

*Η εργασία έχει ως αντικείμενο τη διασύνδεση(interface) ενός PS2 πληκτρολογίου με κάποιο host machine. Αρχικά θα παρουσιαστεί θεωρητικά το πρωτόκολλο αναλύοντας τα ηλεκτρικά και φυσικά χαρακτηριστικά αυτού. Για το ρόλο του host machine έχει επιλεγεί ένα Spartan 3AN FPGA Board της Xilinx. Παρουσιάζεται ο απαιτούμενος κώδικας σε Verilog για την υλοποίηση του PS2 καθώς και όλα τα σχηματικά και λογικά διαγράμματα των απαιτούμενων κυκλωμάτων. Τέλος παρουσιάζονται οι λεπτομέρειες επικοινωνίας ενός PS2 πληκτρολογίου με ένα PC μέσω του FPGA board.*

## **Περιεχόμενα:**

### **Κεφάλαιο 1<sup>ο</sup> – PS2 Keyboard Interface**

1. Εισαγωγή
2. Φυσική Διασύνδεση – Physical Interface
3. Ηλεκτρική Διασύνδεση – Electrical Interface
4. Επικοινωνία: Γενική Περιγραφή
5. Device to Host Επικοινωνία
6. Host to Device Επικοινωνία
7. Κώδικες Σάρωσης – Scan Codes

### **Κεφάλαιο 2<sup>ο</sup> – FPGA – Field Programmable Gate Arrays**

1. Εισαγωγή – Γενικά Χαρακτηριστικά
2. Αρχιτεκτονική
3. Σχεδιασμός – Προγραμματισμός FPGA
4. Spartan 3AN – Σύντομη Περιγραφή Αναπτυξιακής Πλακέτας

### **Κεφάλαιο 3<sup>ο</sup> – Κώδικας Υλοποίησης – Λειτουργικά Modules**

1. PS2 Port Receiving Module
2. PS2 Scan Code Monitor Module
3. PS2 Interface Circuit
4. Universal Asynchronous Receiver Transmitter – UART
  1. Εισαγωγή
  2. UART Receiver
  3. UART Transmitter
  4. UART Overall System
5. Πρόσθετα Αρχεία Κώδικα

### **Κεφάλαιο 4<sup>ο</sup> – Verification Circuits – Κυκλώματα επιβεβαίωσης Ορθής Λειτουργίας**

1. UART Verification Circuit
2. PS2 Verification Circuit

### **Βιβλιογραφία - Πηγές**

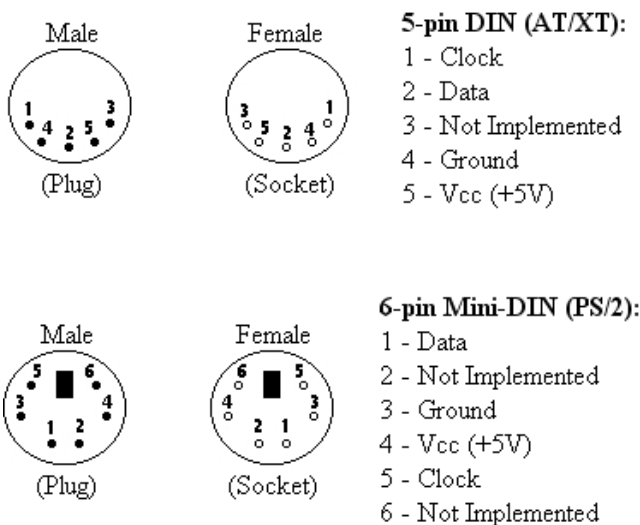
## Κεφάλαιο 1<sup>ο</sup> – PS2 Interface

### 1.1 Εισαγωγή

Το PS2 port παρουσιάστηκε στους προσωπικούς υπολογιστές τις IBM Personal System/2. Πρόκειται για ένα ευρέως διαδεδομένο interface επικοινωνίας τόσο του πληκτρολογίου όσο και του mouse (ποντικιού) με τον εκάστοτε host. Το PS2 port για τους σκοπούς της επικοινωνίας διαθέτει 2 γραμμές, μία γραμμή για την μεταφορά των δεδομένων, τα οποία και μεταφέρονται σειριακά, και μία γραμμή για την μετάδοση της πληροφορίας του ρολογιού (clock), το οποίο καθορίζει πότε τα δεδομένα είναι έγκυρα και μπορούν να διαβαστούν. Η πληροφορία μεταδίδεται σε **πακέτα των 11-bit**, τα οποία περιέχουν: ένα bit έναρξης (**start bit**), 8 bits πληροφορίας (**data bits**), ένα bit περιττής ισοτιμίας (**odd parity bit**) και ένα bit τερματισμού (**stop bit**) (στην host-to-device επικοινωνία υπάρχει και ένα 12<sup>ο</sup> ACK bit). Παρόλο που το βασικό μοτίβο των πακέτων είναι ίδιο για το πληκτρολόγιο και το ποντίκι, υπάρχει διαφορετική ερμηνεία στα bit πληροφορίας. Η επικοινωνία είναι διπλής κατεύθυνσης (bidirectional – host-to-device – device-to-host) και ο host μπορεί να στείλει εντολές στη συσκευή για να ρυθμίσει συγκεκριμένες παραμέτρους.

### 1.2 Φυσική Διασύνδεση – Physical Interface

Το φυσικό PS/2 port έχει 2 μορφών βύσματα σύνδεσης (connectors): Το 5-pin DIN και το 6-pin mini-DIN (βλέπε εικόνα παρακάτω για τη μορφή των βυσμάτων-ακροδεκτών). Και τα δύο είναι ακριβώς (ηλεκτρικά) ίδια, με μόνη πρακτική διαφορά αυτή της οργάνωσης των ακροδεκτών. Αυτό σημαίνει ότι μπορεί να γίνει εύκολα μετάβαση από τον ένα στον άλλο με ένα προσαρμογέα (adaptor). Το DIN πρότυπο δημιουργήθηκε από το Γερμανικό Οργανισμό Προτυποποιήσεων (Deutsch Standardization Organization).

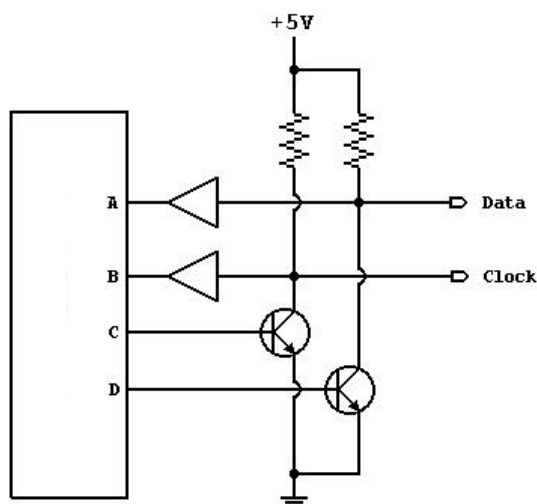


Τα πληκτρολόγια μπορούν να κάνουν χρήση τόσο του 5-pin όσο και του 6-pin προτύπου, αλλά με το δεύτερο να έχει επικρατήσει. Αυτά με 6-pin διασύνδεση είναι ευρέως γνωστά ως “PS2 Πληκτρολόγια”, ενώ αυτά με 5-pin διασύνδεση ονομάζονται “AT” συσκευές (ή “XT” το οποίο χρησιμοποιούνταν αρκετά παλαιότερα). Όλα τα μοντέρνα πληκτρολόγια τα οποία προορίζονται για PCs βασίζονται πλέον στο USB πρωτόκολλο, κάτι το οποίο όμως δεν θα αναλυθεί εδώ σε αυτή την εργασία.

### 1.3 Ηλεκτρική Διασύνδεση – Electrical Interface

$V_{CC}$  (+3.3V - +5.5V) και Ground παρέχουν την απαιτούμενη ισχύ στην συσκευή. Το ρεύμα που αντλεί η συσκευή δεν πρέπει να ξεπερνάει τα 275mA ενώ ειδική μέριμνα πρέπει να υπάρχει για την αποφυγή των αυξομειώσεων στην τάση τροφοδοσίας. Τέτοιες απότομες διακυμάνσεις στην τάση τροφοδοσίας, μπορούν να δημιουργηθούν από το “hot-pluggin” (σύνδεση/αποσύνδεση της συσκευής όσο το σύστημα είναι σε λειτουργία). Για αυτό το λόγο όλες οι μητρικές έχουν τρόπους προστασίας (auto-reset Poly fuses) αλλά καλό είναι να αποφεύγεται από το χρήστη η σύνδεση συσκευών όσο είναι ενεργό το σύστημα.

Τόσο η γραμμή των δεδομένων όσο και αυτή του ρολογιού είναι ανοιχτού συλλέκτη (open-collector) με pullup αντιστάσεις στο  $V_{CC}$ . Μία συνδεσμολογία ανοιχτού συλλέκτη έχει δύο πιθανές καταστάσεις (χαμηλή-low και υψηλής εμπέδωσης-high impedance). Στη χαμηλή κατάσταση, ένα τρανζίστορ άγει προς τη γείωση. Ενώ στην υψηλή εμπέδωση, το σύστημα συμπεριφέρεται ως ένα ανοιχτό κύκλωμα και δεν οδηγεί καθόλου τη γραμμή (ούτε χαμηλό ούτε υψηλό δυναμικό).



*Σημείωση: Στο διπλανό σχήμα έχει γίνει χρήση του ίδιου ακροδέκτη για είσοδο και έξοδο και έχουν χρησιμοποιηθεί οι εσωτερικές pullup αντιστάσεις του host. Στο αναπτυξιακό που παρουσιάζουμε αυτές έχουν 270Ω τιμή.*

### 1.4 Επικοινωνία: Γενική Περιγραφή

Τα PS/2 πληκτρολόγια υλοποιούν ένα σύγχρονο σειριακό πρωτόκολλο διπλής κατεύθυνσης. Η αρτηρία είναι σε κατάσταση “idle” όταν και οι δύο γραμμές είναι σε υψηλό δυναμικό (open-collector). Αυτή είναι η μόνη κατάσταση από την οποία επιτρέπεται η εκκίνηση της μετάδοσης δεδομένων από και προς τη συσκευή. Ο host

έχει πλήρη έλεγχο πάνω στην αρτηρία και μπορεί να εμποδίσει την επικοινωνία οποιαδήποτε χρονική στιγμή θέτοντας τη γραμμή του ρολογιού σε χαμηλό δυναμικό (low).

Εάν ο host θέλει να στείλει δεδομένα στο πληκτρολόγιο, πρέπει πρώτα να διακόψει την επικοινωνία θέτοντας τη γραμμή του ρολογιού σε low. Στη συνέχεια ο host θέτει τη γραμμή δεδομένων σε low και απελευθερώνει τη γραμμή ρολογιού. Αυτή είναι η “Request-to-Send” κατάσταση και ειδοποιεί τη συσκευή για να αρχίσει να παράγει παλμούς ρολογιού (κάθε συσκευή περιέχει μία γεννήτρια παλμών).

Τα παραπάνω συνοψίζονται στον παρακάτω πίνακα:

Data	Clock	Λειτουργία
High	High	Idle
High	Low	Διακοπή Επικοινωνίας
low	high	Host Request-to-send

Όλα τα δεδομένα μεταφέρονται με ένα byte (8 bits) τη φορά και κάθε ένα από αυτά στέλνεται σε πακέτα των 11 bits (ή 12 bits), τα οποία είναι τα εξής:

- 1 ψηφίο εκκίνησης – start bit
- 8 ψηφία δεδομένων – data bits
- 1 ψηφίο περιττής ισοτιμίας – odd parity bit
- 1 ψηφίο τερματισμού – stop bit
- 1 ψηφίο επιβεβαίωσης – ACK bit (μόνο στην host-to-device)

Το ψηφίο ισοτιμίας παίρνει την τιμή 1 εάν υπάρχει ζυγός αριθμός άσων στα ψηφία δεδομένων και την τιμή 0 εάν υπάρχει μονός αριθμός άσων στα ψηφία δεδομένων. Ο αριθμός των άσων στα δεδομένα και το ψηφίο ισοτιμίας πάντα έχουν περιττό άθροισμα. Το ψηφίο ισοτιμίας χρησιμοποιείται για την ανίχνευση των λαθών στην επικοινωνία.

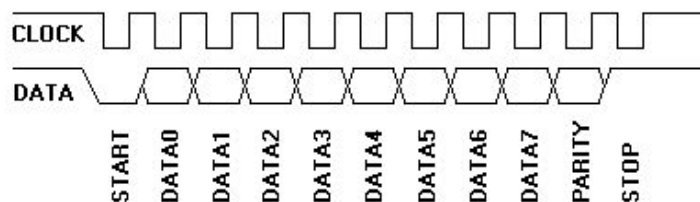
Τα δεδομένα διαβάζονται από τον host κατά την καθοδική ακμή του ρολογιού, ενώ τα δεδομένα που στέλνονται από τον host στη συσκευή διαβάζονται στην ανοδική ακμή του ρολογιού. Η συχνότητα του ρολογιού πρέπει να είναι **μεταξύ 10 και 16,7 kHz**. Αυτό σημαίνει πως το ρολόι πρέπει να είναι σε υψηλό δυναμικό για 30 – 50 μs και σε χαμηλό για άλλο τόσο.

## 1.5 Device to Host Επικοινωνία

Τόσο η γραμμή δεδομένων όσο και αυτή του ρολογιού είναι ανοιχτού συλλέκτη. Μία αντίσταση συνδέεται μεταξύ του της γραμμής και των +5V, άρα η κατάσταση idle της αρτηρίας είναι high. Όταν η συσκευή θέλει να στείλει πληροφορία, πρώτα ελέγχει την γραμμή του ρολογιού για να σιγουρευτεί ότι είναι σε υψηλό δυναμικό. Εάν δεν είναι, σημαίνει ότι ο host έχει εμποδίσει την επικοινωνία και η συσκευή πρέπει να αποθηκεύσει προσωρινά τα δεδομένα σε

κάποιο τοπικό buffer μέχρι να ελευθερώσει το ρολόι ο host. Η γραμμή του ρολογιού πρέπει να είναι συνεχόμενα για τουλάχιστον 50 μs σε υψηλό δυναμικό πριν η συσκευή μπορέσει να μεταφέρει δεδομένα.

Η συσκευή γράφει ένα bit στη γραμμή δεδομένων όταν το ρολόι είναι high και διαβάζεται από τον host όταν το ρολόι είναι low. Η διαδικασία αυτή συνοψίζεται στις παρακάτω εικόνες:



Η παρακάτω εικόνα περιέχει τον κωδικό σάρωσης του πλήκτρου "Q" (15h) ο οποίος αποστέλλεται από το πληκτρολόγιο στον υπολογιστή. Το κανάλι A είναι το σήμα ρολογιού και το κανάλι B το σήμα των δεδομένων.



Η συχνότητα του ρολογιού είναι 10 – 16.7 kHz. Ο χρόνος από την ανοδική ακμή του ρολογιού μέχρι την μετάβαση των δεδομένων πρέπει να είναι το ελάχιστο 5 μs. Ο χρόνος από την μετάβαση των δεδομένων μέχρι την καθοδική ακμή του ρολογιού πρέπει να είναι το ελάχιστον 5 μs και όχι μεγαλύτερο από 25μs.

Ο host μπορεί να διακόψει την επικοινωνία οποιαδήποτε χρονική στιγμή θέτοντας τη γραμμή του ρολογιού σε χαμηλό δυναμικό για τουλάχιστον 100μs. Εάν η επικοινωνία διακοπή πριν από τον 11<sup>ο</sup> παλμό, η συσκευή πρέπει να διακόψει την τρέχουσα μετάδοση και να προετοιμαστεί να ξανά-μεταδώσει αυτή την πληροφορία όταν ο host απελευθερώσει το clock. Για παράδειγμα εάν γίνει η διακοπή κατά την αποστολή του δεύτερου byte από ένα 2-byte break code, πρέπει να μεταδοθούν ξανά και τα δύο bytes του break code και όχι μόνο αυτό που διακόπηκε.

Εάν ο host θέσει το ρολόι σε low πριν από την πρώτη μετάβαση high-to-low, ή μετά την καθοδική ακμή του τελευταίου παλμού ρολογιού, η συσκευή δεν

χρειάζεται να μεταδώσει ξανά κανένα δεδομένο. Παρόλα αυτά, εάν δημιουργηθούν νέα δεδομένα πρέπει να αποθηκευτούν σε buffer μέχρι να απελευθερώσει ο host το ρολόι. Τα πληκτρολόγια έχουν ένα 16-byte buffer για αυτό το σκοπό. Εάν συμβούν περισσότερες από 16-byte πιέσεις πλήκτρων (βλέπε παρ. 1.7), αυτές αγνοούνται μέχρι να υπάρξει χώρος στον buffer.

## 1.6 Host to Device Επικοινωνία

Τα πακέτα τα οποία αποστέλλονται είναι ελάχιστα διαφορετικά στην host-to-device επικοινωνία. Καταρχήν, μία PS/2 συσκευή πάντα παράγει παλμούς ρολογιού. Εάν ο host θέλει να στείλει δεδομένα, πρέπει πρώτα να θέσει τις γραμμές δεδομένων και ρολογιού σε κατάσταση request-to-send ως εξής:

- Διακόπτει την επικοινωνία θέτοντας τη γραμμή ρολογιού σε low για τουλάχιστον 100μs
- Θέτει την γραμμή δεδομένων σε low και απελευθερώνει το ρολόι

Η συσκευή πρέπει να ελέγχει για αυτή την κατάσταση περιοδικά και όχι με χρόνο πάνω από 10ms. Όταν η συσκευή ανιχνεύσει αυτή την κατάσταση, ξεκινάει να παράγει σήμα ρολογιού και δεδομένα σε ομάδες των 8bit ακολουθούμενα από ένα ψηφίο διακοπής (stop bit). Ο host αλλάζει τη γραμμή δεδομένων μόνο όταν η γραμμή ρολογιού είναι σε χαμηλό δυναμικό, και τα δεδομένα διαβάζονται από τη συσκευή όταν το ρολόι είναι σε high. Πρόκειται ουσιαστικά δηλαδή για το αντίθετο από ότι στην device-to-host επικοινωνία.

Μετά από τη λήψη του ψηφίου τερματισμού, η συσκευή επιβεβαιώνει το ληφθέν byte θέτοντας τη γραμμή δεδομένων σε low και παράγοντας ένα τελευταίο παλμό ρολογιού. Εάν ο host δεν απελευθερώσει τη γραμμή δεδομένων μετά των 11<sup>ο</sup> παλμό, η συσκευή θα συνεχίσει να παράγει παλμούς έως ότου η γραμμή δεδομένων απελευθερωθεί (η συσκευή θα παράγει σφάλμα).

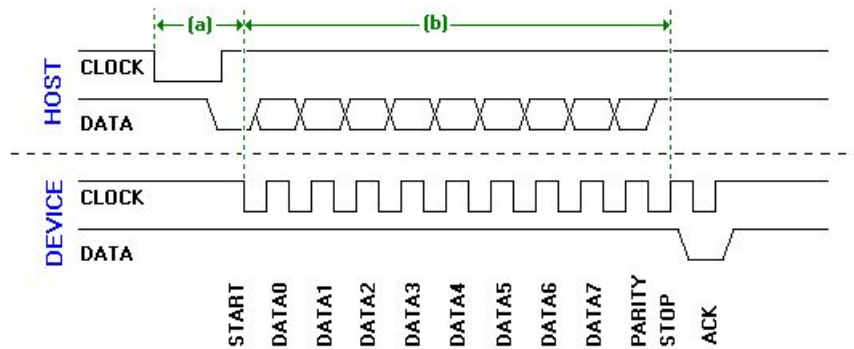
Ο host μπορεί να διακόψει την επικοινωνία σε χρόνο μικρότερο από τον 11<sup>ο</sup> παλμό (ack bit) κρατώντας τη γραμμή του ρολογιού σε χαμηλό δυναμικό για τουλάχιστον 100μs.

Συνοψίζουμε λοιπόν την διαδικασία στα παρακάτω 11 βήματα:

1. Η γραμμή ρολογιού σε low για τουλάχιστον 100μs
2. Γραμμή δεδομένων σε low
3. Απελευθέρωση της γραμμής ρολογιού
4. Αναμονή προκειμένου η συσκευή να φέρει τη γραμμή ρολογιού σε low
5. Αποστολή του πρώτου ψηφίου δεδομένων
6. Αναμονή προκειμένου η συσκευή να θέσει το ρολόι σε high
7. Αναμονή προκειμένου η συσκευή να θέσει το ρολόι σε low
8. Επανάληψη των βημάτων 5-7 για ακόμα 7 ψηφία δεδομένων και το ψηφίο ισοτιμίας
9. Απελευθέρωση της γραμμής δεδομένων
10. Αναμονή μέχρι να θέσει η συσκευή τη γραμμή δεδομένων σε low

11. Αναμονή μέχρι να θέσει η συσκευή τη γραμμή ρολογιού σε low
12. Αναμονή μέχρι η συσκευή να απελευθερώσει τις γραμμές δεδομένων και ρολογιού

Η επόμενη εικόνα παρουσιάζει γραφικά όσα αναλύθηκαν παραπάνω:

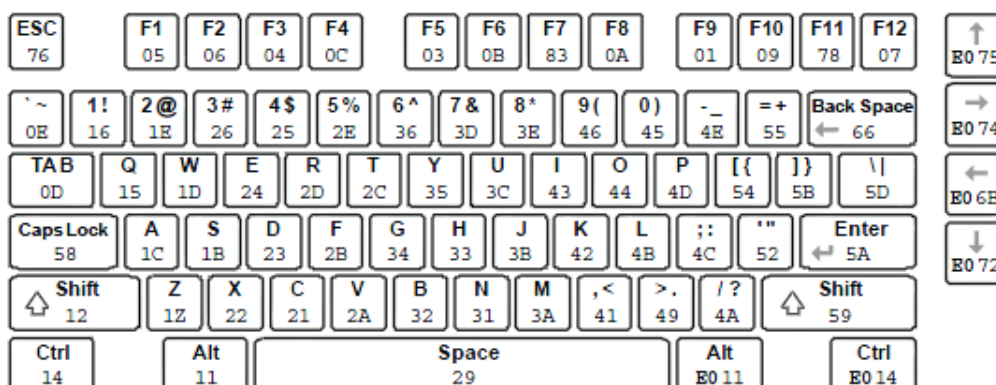


## 1.7 Κώδικες Σάρωσης – Scan Codes

Ένα πληκτρολόγιο αποτελείται από ένα μητρώο πλήκτρων και ένα ενσωματωμένο μικροελεγκτή (embedded microcontroller) ο οποίος σαρώνει (scans) τις δραστηριότητες των πλήκτρων και στέλνει τους ανάλογους κωδικούς σάρωσης (scan codes). Υπάρχουν 3 ειδών δραστηριότητες στα πλήκτρα:

- Όταν ένα πλήκτρο πιέζεται, ο ανάλογος κωδικός του πλήκτρου (make code) μεταδίδεται
- Όταν ένα πλήκτρο πιέζεται συνεχόμενα, μία κατάσταση γνωστή και ως typematic, ο κωδικός του πλήκτρου μεταδίδεται συνεχώς με μία συγκεκριμένη ταχύτητα. Εξ ορισμού, ένα PS2 πληκτρολόγιο μεταδίδει τους κωδικούς περίπου κάθε 100ms μετά την πίεση του πλήκτρου για 0.5sec.
- Όταν ένα πλήκτρο απελευθερώνεται, ο κωδικός διακοπής (break code) μεταδίδεται.

Οι κωδικοί των πλήκτρων του κυρίως κομματιού του PS2 πληκτρολογίου (κάθε εταιρία κατασκευής ενδέχεται να προσθέτει δικές τις λειτουργίες) φαίνεται στην παρακάτω εικόνα.





Ο κωδικός είναι συνήθως μήκους 1 byte και αναπαρίσταται από δύο δεκαεξαδικά (hex) ψηφία. Για παράδειγμα ο κωδικός του πλήκτρου A είναι 1C. Αυτός ο κωδικός μπορεί να εισαχθεί σε ένα πακέτο και να αποσταλεί. Ο κωδικός από μερικά πλήκτρα ειδικού σκοπού, τα οποία είναι γνωστά και ως πλήκτρα επέκτασης (extended keys), μπορούν να έχουν μήκος 2 έως 4 byte. Για παράδειγμα ο κωδικός του άνω βέλους είναι E0 75. Συνεπώς πολλαπλά πακέτα χρειάζονται για αυτές τις μεταφορές πληροφορίας. Οι κωδικοί διακοπής των συνηθισμένων πλήκτρων αποτελούνται από το F0<sub>h</sub> ακολουθούμενο από τον κωδικό του πλήκτρου. Για παράδειγμα ο κωδικός διακοπής του A είναι F0 1C.

Το PS2 πληκτρολόγιο μεταδίδει μία ακολουθία από κωδικούς ανάλογα με την δραστηριότητα των πλήκτρων. Για παράδειγμα όταν πιέζουμε και απελευθερώνουμε το πλήκτρο A, το πληκτρολόγιο μεταδίδει τον κωδικό του πλήκτρου και τον κωδικό διακοπής, 1C F0 1C.

Εάν το πλήκτρο πατηθεί για αρκετή ώρα προτού απελευθερωθεί, τότε ο κωδικό μεταδίδεται πολλαπλές φορές (1C 1C 1C ... 1C F0 1C).

Πολλαπλά πλήκτρα μπορούν να πατηθούν την ίδια χρονική στιγμή. Για παράδειγμα πρώτα μπορεί να πατηθεί το πλήκτρο shift (του οποίου ο κωδικός είναι 12) και μετά να πατηθεί το A, έπειτα να απελευθερωθεί το A και τέλος να απελευθερωθεί και το shift. Ο αποστελλόμενος κωδικός ακολουθεί τους κωδικούς σάρωσης και τους κωδικούς διακοπής των 2 πλήκτρων (12 1C F0 1C F0 12).

Τέλος έχουμε και τις εντολές που μπορεί να στείλει ο host στο PS2 πληκτρολόγιο. Οι κυριότερες από αυτές παρουσιάζονται στον παρακάτω πίνακα:

Κωδικός	Περιγραφή
ED	Turn on/off Num Lock, Caps Lock και Scroll Lock LEDs. Αποστέλλεται ένα πακέτο στο οποίο εάν υπάρχει 1 στα bit 0,1 και 2 (Scroll, Num, Caps Lock αντίστοιχα) ανάβει το αντίστοιχο LED (τα υπόλοιπα bit αγνοούνται)
EE	Echo – Το πληκτρολόγιο απαντάει με τον ίδιο κωδικό
F3	Set scan code repeat rate – Το πληκτρολόγιο επιβεβαιώνει την λήψη του F3 επιστρέφοντας ένα FA, μετά το οποία ο host στέλνει ένα δεύτερο byte για να καθορίσει το ρυθμό επανάληψης
FE	Resend – Επαναποστολή του τελευταίου κωδικού
FF	Reset

## Κεφάλαιο 2<sup>ο</sup> – FPGA (Field Programmable Gate Arrays)

### 2.1 Εισαγωγή – Γενικά Χαρακτηριστικά

Το FPGA (Field Programmable Gate Array – Λογικές Διατάξεις Πυλών Προγραμματιζόμενου Πεδίου) είναι ένα ολοκληρωμένο κύκλωμα σχεδιασμένο να προγραμματίζεται από τον καταναλωτή ή τον σχεδιαστή μετά την κατασκευή του. Ο προγραμματισμός του πραγματοποιείται κυρίως με χρήση μίας γλώσσας

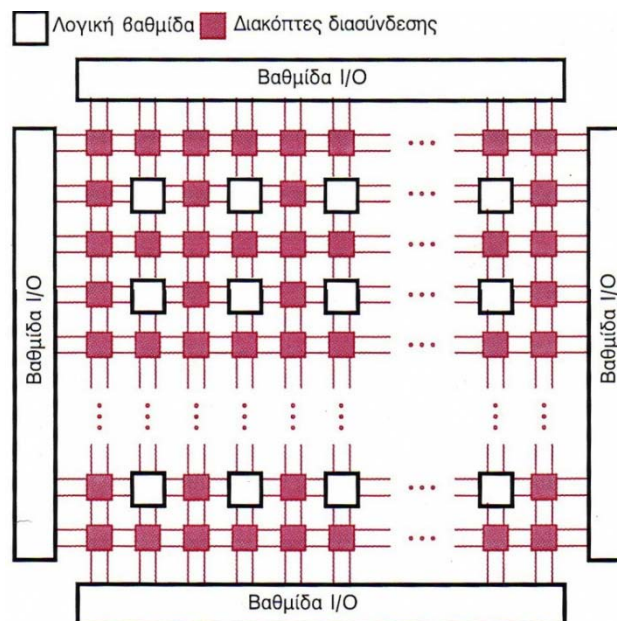
περιγραφής υλικού (HDL – Hardware Description Language), παρόμοια με αυτή που χρησιμοποιούνταν για τα ASIC (Application Specific Integrated Circuits). Τα FPGA μπορούν να χρησιμοποιηθούν για να υλοποιήσουν οποιαδήποτε λογική συνάρτηση την οποία ένα ASIC μπορεί να εκτελέσει. Η δυνατότητα αναβάθμισης της λειτουργικότητας μετά την αγορά, και το χαμηλό κόστος σε σχέση με ένα ASIC σχεδιασμό, προσφέρει πλεονεκτήματα για αρκετές εφαρμογές.

Τα FPGA περιέχουν αντικείμενα προγραμματιζόμενης λογικής, τα λεγόμενα *block cells* (λογική βαθμίδα), και μία ιεραρχία από επανα-προγραμματιζόμενες εσωτερικές διασυνδέσεις (interconnects), οι οποίες επιτρέπουν στα block να διασυνδέονται φυσικά μεταξύ τους. Τα logic blocks μπορούν να προγραμματιστούν για να εκτελέσουν πολύπλοκες συνδυαστικές συναρτήσεις, ή πιο απλές λογικές πύλες όπως το λογικό ΚΑΙ (AND). Στα περισσότερα FPGA τα logic blocks περιέχουν επίσης και στοιχεία μνήμης, τα οποία μπορεί να είναι απλά flip-flops ή ολόκληρα block μνήμης.

Εφαρμογές των FPGA αποτελούν οι εξής: ψηφιακή επεξεργασία σήματος, αεροδιαστημική, αμυντικά συστήματα, προτυποποίηση ASIC, φαρμακευτική, αναγνώριση φωνής, κρυπτογραφία, βιοιατρική, εξομοίωση υλικού, αστρονομία, ανίχνευση μετάλλων και ένα μεγάλο ακόμα φάσμα τομέων.

## 2.2 Αρχιτεκτονική

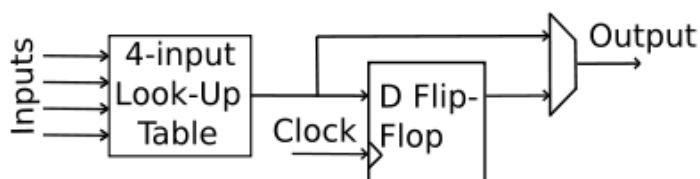
Η γενική δομή των FPGA περιέχει 3 ειδών components: λογικές βαθμίδες, βαθμίδες εισόδου/εξόδου για τη σύνδεση με τους ακροδέκτες της συσκευασίας και διακόπτες και γραμμές εσωτερικής διασύνδεσης. Αυτά φαίνονται στην παρακάτω εικόνα:



Οι λογικές βαθμίδες οργανώνονται με τη μορφή δισδιάστατης σειράς και οι γραμμές διασύνδεσης οργανώνονται ως οριζόντια και κατακόρυφα κανάλια δρομολόγησης ανάμεσα στις γραμμές και τις στήλες των λογικών βαθμίδων. Τα κανάλια αυτά εμπεριέχουν καλώδια και προγραμματιζόμενους διακόπτες που επιτρέπουν στις λογικές βαθμίδες να διασυνδέονται με πολλούς τρόπους.

Παρατηρώντας το παραπάνω σχήμα βλέπουμε ότι υπάρχουν 2 θέσεις προγραμματιζόμενων διακοπών: Τα τετράγωνα που βρίσκονται δίπλα στις λογικές βαθμίδες (πάνω, κάτω, δεξιά και αριστερά) περιέχουν διακόπτες που συνδέουν τους ακροδέκτες εισόδου και εξόδου των λογικών μονάδων με τα καλώδια διασύνδεσης και τα τετράγωνα που βρίσκονται διαγώνια μεταξύ των λογικών βαθμίδων συνδέουν ένα καλώδιο διασύνδεσης με ένα άλλο (πχ ένα οριζόντιο με ένα κατακόρυφο). Επίσης υπάρχουν προγραμματιζόμενες συνδέσεις ανάμεσα στις βαθμίδες εισόδου/εξόδου και τα καλώδια διασύνδεσης. Ο αριθμός των καλωδίων και των προγραμματιζόμενων διακοπών ποικίλει στα διάφορα ολοκληρωμένα του εμπορίου.

Μία κλασική λογική βαθμίδα αποτελείται από ένα πίνακα αντιστοίχισης 4 εισόδων (4-input lookup table-LUT) και ένα flip-flop, όπως αναπαρίσταται από την παρακάτω εικόνα:



Υπάρχει μόνο μία έξοδος, ή οποία μπορεί να είναι είτε η εκταμιευμένη (registered) είτε ή μη εκταμιευμένη (unregistered) έξοδος του LUT. Η λογική βαθμίδα έχει 4 εισόδους για τον πίνακα και μία είσοδο ρολογιού. Τα σήματα ρολογιού (και αρκετές φορές και αρκετά άλλα high-fanout σήματα) συνήθως δρομολογούνται μέσα από ειδικά δίκτυα δρομολόγησης (DCM – Digital Clock Management).

### 2.3 Σχεδιασμός – Προγραμματισμός FPGA

Ο προγραμματισμός των FPGA γίνεται μέσα από μία γλώσσα περιγραφής υλικού (hardware description language – HDL) ή από μία σχηματική σχεδίαση (schematic design). Μία HDL γλώσσα είναι πιο κατάλληλη για εργασία με μεγάλες και απαιτητικές υλοποιήσεις, επειδή είναι δυνατόν να προσδιοριστούν επακριβώς τα διακριτά modules χωρίς να χρειάζεται να γίνεται σχεδιασμός με το χέρι. Παρόλα αυτά μία σχηματική προσέγγιση παρέχει μία οπτική επαφή της υλοποίησης.

Στη συνέχεια χρησιμοποιώντας ένα εργαλείο ηλεκτρονικού σχεδιασμού αυτοματισμού (electronic design automation) παράγεται μία mapped netlist. Αυτή η netlist μπορεί στη συνέχεια να προσαρμοστεί στην FPGA αρχιτεκτονική μέσω μιας διαδικασίας που ονομάζεται place-and-route, η οποία συνήθως πραγματοποιείται από ανάλογο συνοδευτικό λογισμικό της εταιρίας παραγωγής του FPGA. Ο χρήστης

μπορεί να αξιολογήσει τη χαρτογράφηση, την τοποθέτηση και την δρομολόγηση των αποτελεσμάτων μέσα από χρονικές αναλύσεις, εξομοίωση και άλλες τεχνικές εγκυρότητας. Όταν ο σχεδιασμός και η εγκυρότητα (validation) έχουν επιτευχθεί, παράγεται το δυαδικό (binary) αρχείο, το οποίο χρησιμοποιείται για να (επανά)παραμετροποιήσει το FPGA (πάλι μέσω του επίσημου λογισμικού της εταιρίας). Το binary αρχείο μεταφέρεται στο FPGA μέσα από μία σειριακή διασύνδεση (serial interface JTAG) ή σε μία εξωτερική μνήμη συνήθως μία EEPROM (και πιο πρόσφατα και FLASH).

Η πιο διαδεδομένες HDL γλώσσες είναι η VHDL και η Verilog, παρόλα αυτά γίνεται μία προσπάθεια να μειωθεί η πολυπλοκότητα του σχεδιασμού μέσα από HDL γλώσσες, οι οποίες συγκρίνονται με τις αντίστοιχες γλώσσες μηχανής (assembly languages). Υπάρχουν προσπάθειες για να δημιουργηθεί ένα είδος αφαιρετικότητας (abstraction) στα διάφορα επίπεδα μέσα από την εισαγωγή εναλλακτικών γλωσσών. Επίσης για ακόμα μεγαλύτερη ευελιξία, ο σχεδιασμός μπορεί να γίνει και μέσα από microcontroller software cores (βλέπε Xilinx Picoblaze), για την ανάπτυξη απλών εφαρμογών όχι και τόσο κρίσιμου χρόνου (κύριος δηλαδή για εκπαιδευτικούς, αναπτυξιακούς και ερασιτεχνικούς σκοπούς).

Η γλώσσα που χρησιμοποιήθηκε για αυτό το project είναι η Verilog στη νέα της έκδοσης 2001 (υπάρχει και 2005 έκδοση αλλά οι διαφορές είναι ελάχιστες και δεν επηρεάζουν την παρούσα εργασία).

## 2.4 Spartan 3AN – Σύντομη Περιγραφή Αναπτυξιακής Πλακέτας



Σύνοψη των χαρακτηριστικών:

- Xilinx 700K-gate XC3S700AN Spartan 3AN nonvolatile FPGA (484-ball BGA)
  - Internal 8Mbit In-System Flash memory
  - Store FPGA configuration bitstream or nonvolatile data
- 4 Mbit Xilinx Platform Flash Configuration PROM
- 64 Mbyte (512Mbit) DDR2 SDRAM, 32Mx16 data interface
- 4 Mbyte (32 Mbit) parallel NOR Flash
  - FPGA configuration storage
  - MicroBlaze code storage
  - x8 or x9 data interface after configuration
- Two 16 Mbit SPI serial Flash
  - STMicroelectronics and Atmel DataFlash serial architectures
  - FPGA configuration storage
  - Nonvolatile data storage
  - MicroBlaze code shadowing
- Two-line 16-character LCD
- PS/2 Port (mouse, keyboard and Y-splitter)
- VGA display port 12-bit color
- 10/100 Ethernet PHY (requires MAC in FPGA)
- On-board USB-based programming solution
- 50 MHz clock oscillator
- SMA connector for clock inputs or outputs
- 100-pin Hirose FX2 expansion connector with up to 43 FPGA I/O pins
- High-speed differential I/O connectors
- Two six-pin expansion connectors for Peripheral Modules
- Four-output, SPI-based DAC (Digital to Analog converter)
- Two-input, SPI-based ADC (Analog to Digital Converter) with programmable pre-amplifier
- Stereo audio jack using digital I/O pins
- ChipScope debugging port
- Rotary-encoder with push-button shaft
- Eight discrete LEDs
- Four slide switches
- Four push-button switches

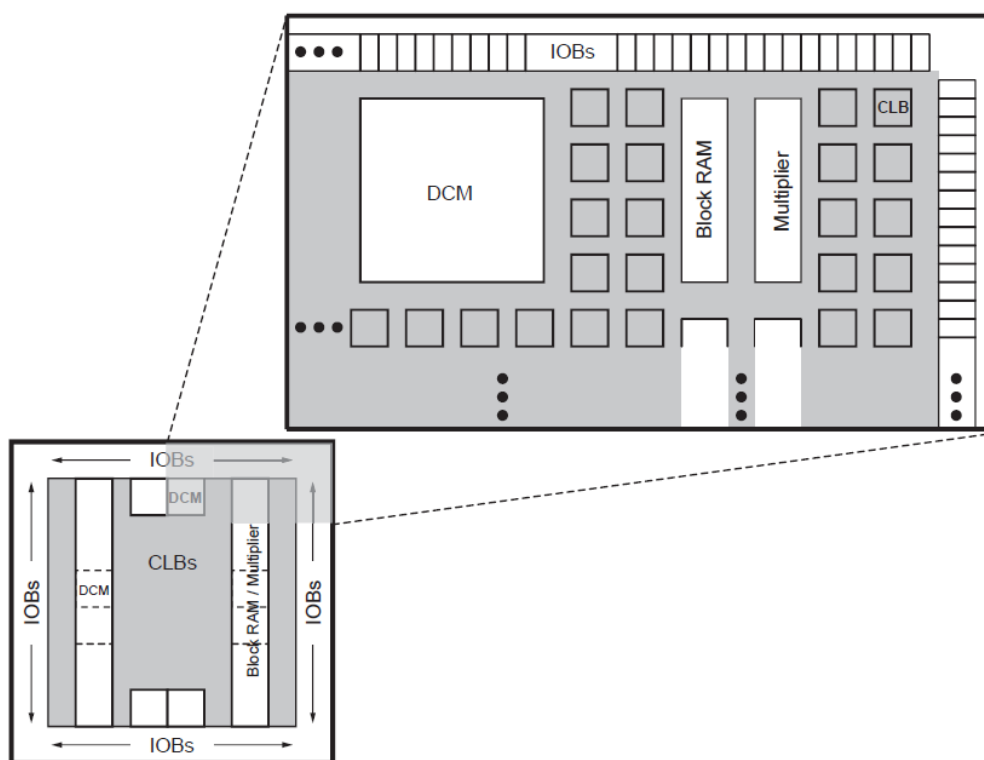
Ακολουθεί και ένας πίνακας με τα βασικά δομικά χαρακτηριστικά όλων των μοντέλων της ίδιας οικογένειας:

Device	System Gates	Equivalent Logic Cells	CLBs	Slices	Distributed RAM bits <sup>(1)</sup>	Block RAM bits <sup>(1)</sup>	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs	Bitstream Size	In-System Flash bits
XC3S50AN	50K	1,584	176	704	11K	54K	3	2	108	50	427K	1M
XC3S200AN	200K	4,032	448	1792	28K	288K	16	4	195	90	1,168K	4M
XC3S400AN	400K	8,064	896	3,584	56K	360K	20	4	311	142	1,842K	4M
XC3S700AN	700K	13,248	1472	5,888	92K	360K	20	8	372	165	2,669K	8M
XC3S1400AN	1400K	25,344	2816	11,264	176K	576K	32	8	502	227	4,644K	16M

Η αρχιτεκτονική των Spartan 3AN FPGA αποτελείται από 5 θεμελιώδη προγραμματιζόμενα λειτουργικά στοιχεία:

- 1) **Configurable Logic Blocks - CLBs(Λογικές Κυψελίδες):** Περιέχουν ευέλικτους πίνακες αντιστοίχισης (Lookup Tables – LUTs) οι οποίοι υλοποιούν λογικά και αποθηκευτικά στοιχεία σαν τα FF(flip-flops) και τους μανδαλωτές (latches).
- 2) **Input/Output Blocks (IOBs):** Ελέγχουν την ροή των δεδομένων μεταξύ των I/O ακροδεκτών και την εσωτερική λογική της συσκευής. Υποστηρίζουν ροή δεδομένων διπλής κατεύθυνσης (bidirectional) και λειτουργία 3-καταστάσεων.
- 3) **Block RAM:** Παρέχουν αποθηκευτικό χώρο με τη μορφή 18-Kbit dual-port blocks.
- 4) **Multiplier Blocks:** Δέχονται δύο 18-bit δυαδικούς αριθμούς σαν εισόδους και υπολογίζουν το αποτέλεσμα
- 5) **Digital Clock Manager (DCM) Blocks:** Παρέχουν αυτό-καλιμπραριζόμενες (self-calibrating), πλήρως ψηφιακές λύσεις για την διανομή, καθυστέρηση, πολλαπλασιασμό, διαίρεση και ολίσθηση-φάσης των σημάτων ρολογιού.

Τα παραπάνω δομικά στοιχεία οργανώνονται ως εξής:



Για περισσότερες λεπτομέρειες σχετικά με τα τεχνικά χαρακτηριστικά ο ενδιαφερόμενος μπορεί να ανατρέξει στα ανάλογα data sheets και user guides (υπάρχει σχετική αναφορά στο τέλος).

## Κεφάλαιο 3<sup>ο</sup> – Κώδικας Υλοποίησης – Λειτουργικά Modules

### 3.1 Port Receiving Module

Όπως αναφέρθηκε στην παράγραφο επεξήγησης του PS2 πρωτοκόλλου έχουμε δύο σήματα: τη γραμμή δεδομένων (*ps2d*) και τη γραμμή ρολογιού (*ps2c*). Τα δεδομένα μεταφέρονται σειριακά και δειγματοληπτούνται κατά την καθοδική ακμή του ρολογιού στην device-to-host επικοινωνία (που μας ενδιαφέρει κυρίως).

Το Port Receiving υποσύστημα περιέχει ένα κύκλωμα αναγνώρισης της καθοδικής ακμής, το οποίο παράγει ένα clock-cycle tick (κύκλο ρολογιού) κατά την καθοδική ακμή του *ps2c* σήματος, και τον αποδέκτη, ο οποίος κάνει shift και συνθέτει τα εισερχόμενα σειριακά δεδομένα.

Λόγο του πιθανού θορύβου και της αργής μετάβασης (transition), εισάγουμε ένα απλό κύκλωμα φιλτραρίσματος για να εξαλείψουμε τα glitches. Ο κώδικας του φίλτρου έχει ως εξής:

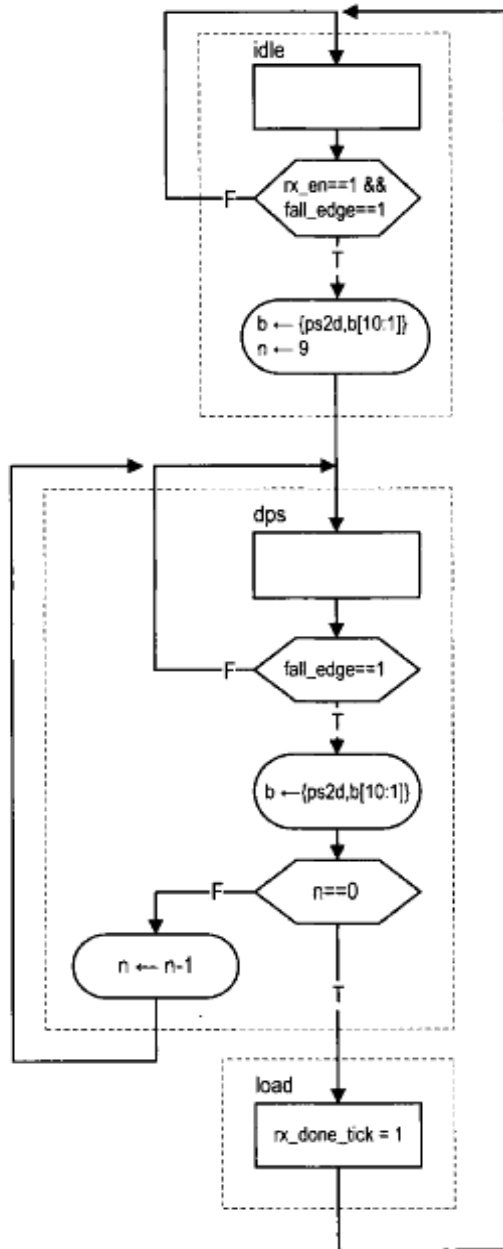
```
always @(posedge clk, posedge reset)
    ...
    filter_reg <= filter_next;
    ...
//1-bit shifter
assign filter_next = {ps2c, filter_reg[7:1]};
// filter
assign f_ps2c_next = (filter_reg==8'b11111111) ? 1'b1 :
                    (filter_reg==8'b00000000) ? 1'b0 :
                    f_ps2c_reg;
```

Το κύκλωμα αποτελείται από έναν 8-bit shift register και επιστρέφει 1 ή 0 όταν 8 συνεχόμενοι άσσοι ή μηδενικά λαμβάνονται. Glitches μικρότερα των 8 κύκλων ρολογιού αγνοούνται (φιλτράρονται). Η φιλτραρισμένη έξοδος στη συνέχεια προωθείται στο κανονικό κύκλωμα ανίχνευσης καθοδικής ακμής (falling edge detection circuit).

*Σημ:* Στην υλοποίηση δεν περιέχεται κύκλωμα ανίχνευσης λαθών μέσα από το parity bit και ούτε έχει εισαχθεί watchdog timer για την αποφυγή εγκλωβισμού σε μία μη αποδεκτή κατάσταση.

- κώδικας του module του receiver περιέχεται στο αρχείο *ps2\_rx.v*

Στην παρακάτω εικόνα παρουσιάζεται το flowchart του υπομονάδας του αποδέκτη (receiver):



Αρχικά ο receiver είναι σε κατάσταση idle. Περιέχει ένα πρόσθετο σήμα ελέγχου,  $rx\_en$ , το οποίο χρησιμοποιείται για να ενεργοποιήσει ή να απενεργοποιήσει την λειτουργία της λήψης. Ο σκοπός του σήματος είναι να συντονίζει την bidirectional επικοινωνία.

Εφόσον το λογικό ΚΑΙ του tick (από το κύκλωμα ανίχνευσης καθοδικής ακμής) και του  $rx\_en$  είναι 1, το start bit γίνεται shift και μεταβαίνουμε στην κατάσταση dps (data processing state). Εφόσον τα ληφθέντα δεδομένα είναι σε προκαθορισμένο format, κάνουμε shift τα εναπομείναντα 10 bits σε μία κατάσταση (αντί να κάνουμε χρήση περισσότερων καταστάσεων – data, parity, stop states).

Έπειτα το διάγραμμα μεταβαίνει στην κατάσταση load, στην οποία ένας extra κύκλος ρολογιού παρέχεται προκειμένου να ολοκληρωθεί και η ολίσθηση του stop bit, και το  $rx\_done\_tick$  σήμα ενεργοποιείται για ένα κύκλο ρολογιού.

### 3.2 Scan Code Monitor Module

Όπως αναφέραμε και την επεξήγηση του πρωτοκόλλου, τα δεδομένα αποστέλλονται με κάποιους κωδικούς. Συνεπώς χρειαζόμαστε ένα σύστημα το οποίο θα παρακολουθεί όλους αυτούς του κωδικούς και θα εκτελεί της ανάλογες ενέργειες. Το Scan Code Monitor κύκλωμα της υλοποίησης κάνει monitor των ληφθέν πακέτων κωδικών και απεικονίζει τους scan codes σε ένα HyperTerminal παράθυρο μέσω σειριακής επικοινωνίας με το PC.

Αρχικά ο ληφθέν κωδικός σάρωσης διαχωρίζεται σε δύο 4-bit κομμάτια και διαχειρίζεται σαν δύο δεκαεξαδικά (hex) ψηφία. Στη συνέχεια μετατρέπονται τα δύο hex ψηφία στην αντίστοιχη ASCII αναπαράσταση και αποστέλλονται στο PC

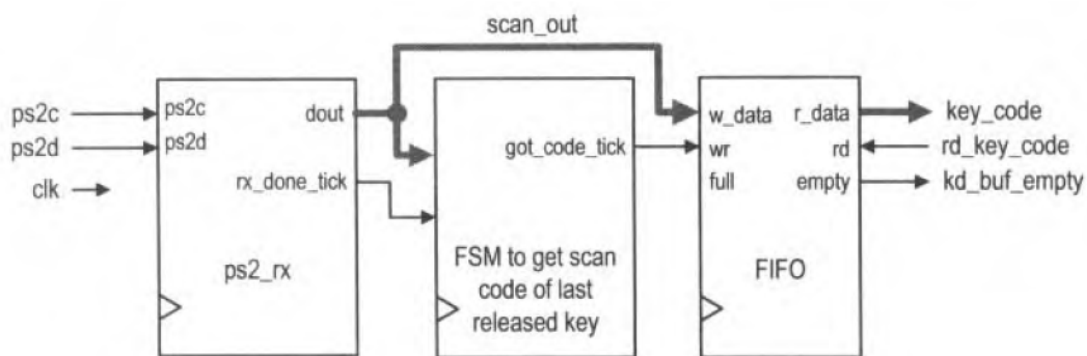


μέσο της UART (Universal Asynchronous Receiver Transmitter). Για την UART υπάρχει περαιτέρω ανάλυση σε επόμενη ενότητα.

- Ο κώδικας του Scan Code Monitor Module περιέχεται στο αρχείο *kb\_monitor.v*

### 3.3 Interface Circuit – Κύκλωμα Διασύνδεσης

Το κύκλωμα του πληκτρολογίου είναι ένα περιφερειακό κύκλωμα ενός μεγάλου συστήματος και συνεπώς χρειάζεται ένα μηχανισμό για να επικοινωνεί με το κυρίως σύστημα. Για το σκοπό αυτό χρησιμοποιούμε ένα four-word buffer για την διασύνδεση στο σχεδιασμό μας. Το top-level block διάγραμμα φαίνεται παρακάτω:



Αποτελείται από τον PS2 receiver, ένα FIFO buffer και ένα FSM ελέγχου. Η βασική ιδέα είναι να γίνει χρήση του FSM για να παρακολουθούνται τα FO πακέτα των break code. Μετά τη λήψη του, το επόμενο πακέτο πρέπει να είναι να είναι ο κωδικός (make code) του πλήκτρου, ο οποίος και γράφεται στον buffer.

- Ο κώδικας του interface circuit περιέχεται στο αρχείο *kb\_code.v*

## 3.4 UART

### 3.4.1 Εισαγωγή

Επειδή το θεωρητικό υπόβαθρο για την UART δεν αποτελεί σκοπό της εργασίας αλλά μέσω για την υλοποίηση και τον έλεγχο λειτουργίας θα παρουσιαστεί εδώ περιληπτικά, με έμφαση στα σημεία που μας ενδιαφέρουν.

Πρόκειται για ένα κύκλωμα το οποίο στέλνει παράλληλα δεδομένα μέσα από μία σειριακή γραμμή. Οι UART χρησιμοποιούνται σε συνδυασμό με το EIA RS-232 πρότυπο, το οποίο ορίζει τα ηλεκτρικά, μηχανικά, λειτουργικά και διαδικαστικά χαρακτηριστικά του απαιτούμενου εξοπλισμού επικοινωνίας και από τις δύο πλευρές. Επειδή η τάση λειτουργίας που ορίζεται στο RS-232 είναι διαφορετική από

αυτή των I/O Pins του FPGA, ένας μετατροπέας τάσης παρεμβάλλεται μεταξύ της RS-232 πόρτας και των ακροδεκτών του FPGA.

Ένα UART περιέχει κυκλώματα αποστολής και παραλαβής. Το κύκλωμα αποστολέα (transmitter) είναι στην ουσία ένας ειδικός καταχωρητής ολίσθησης (shift register) ο οποίος φορτώνει παράλληλα δεδομένα και τα κάνει shift στην έξοδο bit προς bit με ένα συγκεκριμένο ρυθμό (rate). Το κύκλωμα παραλήπτη (receiver) κάνει το αντίθετο, παίρνοντας σαν είσοδο bit προς bit τα δεδομένα και ανασυνθέτει την πληροφορία. Η σειριακή γραμμή είναι σε high όταν είναι σε αναμονή (idle). Η μεταφορά ξεκινάει με ένα start bit , το οποίο είναι 0, ακολουθούμενο από τα bit δεδομένων και ένα προαιρετικό ψηφίο ισοτιμίας, και τελειώνει με stop ψηφία τα οποία είναι σε high. Ο αριθμός των bit δεδομένων είναι 6, 7 ή 8. Ο αριθμός των stop bits μπορεί να είναι 1, 1.5 ή 2. Επίσης σημαντικό είναι ότι το λιγότερο σημαντικό ψηφίο (LSB) είναι αυτό που μεταδίδεται πρώτα.

Κανένα είδος ρολογιού δεν μεταφέρεται μέσα από την σειριακή γραμμή. Πριν από την εκκίνηση της μεταφοράς, ο αποστολέας και ο παραλήπτης πρέπει να συναινέσουν εξ αρχής σε κοινές ρυθμίσεις. Αυτές οι ρυθμίσεις είναι το baud rate (ρυθμός μεταφοράς), ο αριθμός των ψηφίων δεδομένων και των stop bits, και η χρήση ή όχι ψηφίου ισοτιμίας για έλεγχο. Οι πιο ευρέως χρησιμοποιούμενες τιμές για το baud rate είναι 2400, 4800, 9600 και 19,200.

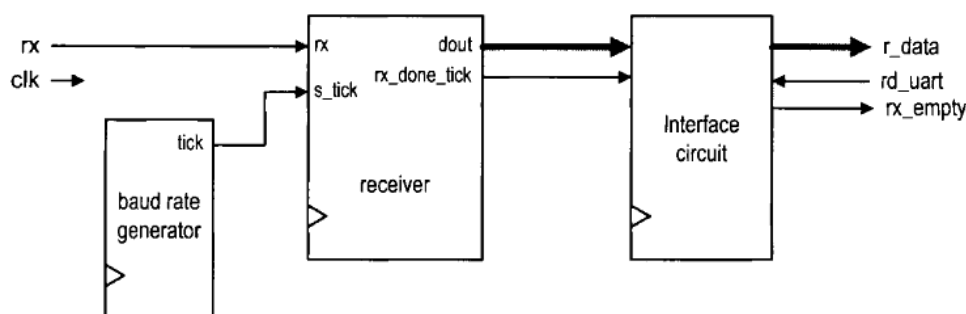
### 3.4.2 Υποσύστημα Παραλήπτη – UART Receiver

Εφόσον δεν υπάρχει ρολόι για το συντονισμό, ο παραλήπτης μπορεί να διαβάσει τα δεδομένα μόνο με χρήση κάποιων προκαθορισμένων παραμέτρων. Χρησιμοποιείται ένα oversampling scheme το οποίο κάνει εκτίμηση των μέσων των μεταδιδόμενων ψηφίων δεδομένων και στη συνέχεια διαβάζει αυτά, στα ανάλογα σημεία.

Ο πιο συνηθισμένος ρυθμός δειγματοληψίας είναι 16 φορές το baud rate, το οποίο σημαίνει ότι κάθε σειριακό bit δειγματοληπτείτε 16 φορές. Ας υποθέσουμε ότι σε μία επικοινωνία έχουμε N ψηφία δεδομένων και M stop bits. Το oversampling scheme θα λειτουργήσει ως εξής:

1. Αναμονή μέχρι το εισερχόμενο σήμα να γίνει 0, η αρχή του start bit, και τότε γίνεται εκκίνηση του sampling tick μετρητή (counter)
2. Όταν ο μετρητής φτάσει στην τιμή 7, το εισερχόμενο σήμα έχει φτάσει στο μεσαίο σημείο του start bit. Καθαρισμός του μετρητή.
3. Όταν ο μετρητής φτάσει στην τιμή 15, το εισερχόμενο σήμα επεξεργάζεται για ένα ψηφίο και φτάνει στο μέσο του ψηφίου. Γίνεται ανάγνωση της τιμής του και shift αυτής σε έναν καταχωρητή, και επανεκκίνηση του μετρητή.
4. Επανάληψη του βήματος 3 για N-1 φορές προκειμένου να διαβαστούν όλα τα ψηφία δεδομένων
5. Εάν υπάρχει parity bit γίνεται επανάληψη του 3 ακόμα μία φορά
6. Επανάληψη του βήματος 3 για ακόμα M φορές για τα stop bits

Το block διάγραμμα υποσυστήματος του παραλήπτη είναι το εξής:



Αποτελείται από 3 κυρίως components:

- UART Receiver: Το κύκλωμα το οποίο διαβάζει τα δεδομένα μέσω oversampling
- Baud rate generator: Το κύκλωμα για την παραγωγή των sampling ticks
- Interface circuit: Το κύκλωμα το οποίο παρέχει ένα buffer και παρεμβάλλεται μεταξύ του UART receiver και του συστήματος που χρησιμοποιεί το UART.

Ο κύριος σκοπός του κυκλώματος διασύνδεσης είναι αρχικά να παρέχει ένα μηχανισμό ειδοποίησης όταν υπάρχει διαθέσιμη νέα λέξη και να αποτρέψει την ληφθέν λέξη από το να διαβαστεί πολλαπλές φορές. Και δεύτερον να παρέχει ένα buffer space (κυρίως flag FF και FIFO buffer) μεταξύ του receiver και του κυρίως συστήματος.

- Ο κώδικας του Receiver υποσυστήματος περιέχεται στο αρχείο *uart\_rx.v*

### 3.4.3 Υποσύστημα Αποστολέα – UART Transmitter

Η οργάνωση του υποσυστήματος του transmitter είναι παρόμοια με αυτή του receiver. Αποτελείται από τα εξής κυκλώματα: UART Transmitter, baud rate generator και interface circuit (κύκλωμα διασύνδεσης). Το κύκλωμα διασύνδεσης είναι παρόμοιο με αυτό του receiver με τη διαφορά ότι το κυρίως σύστημα κάνει set το flag του FF ή γράφει στον FIFO Buffer και ο UART transmitter κάνει clear τα παραπάνω.

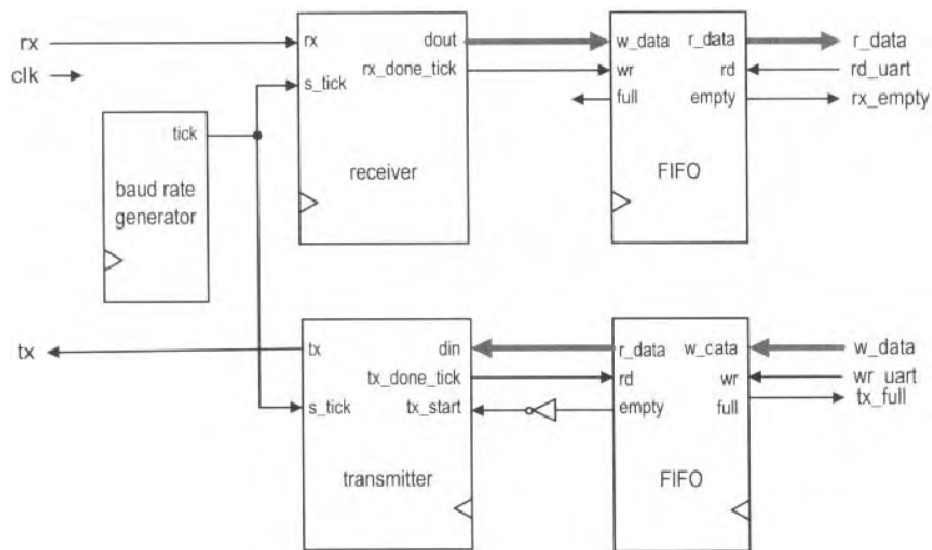
Ο UART transmitter είναι στην ουσία ένας καταχωρητής ολίσθησης (shift register), ο οποίος κάνει shift out τα δεδομένα με ένα συγκεκριμένο ρυθμό. Ο ρυθμός ελέγχεται από one-clock-cycle enable ticks τα οποία παράγονται από τον baud rate generator. Επειδή δεν εμπλέκεται oversampling, η συχνότητα των ticks είναι 16 φορές πιο αργή από αυτή του UART receiver. Αντί να εισαχθεί νέος

μετρητής, ο UART transmitter συνήθως μοιράζεται την ίδια baud rate γεννήτρια με τον receiver και χρησιμοποιεί ένα εσωτερικό μετρητή για την καταμέτρηση των enable ticks.

- Ο κώδικας του Receiver υποσυστήματος περιέχεται στο αρχείο *uart\_tx.v*

### 3.4.4 Συνολικό Σύστημα UART – UART Overall System

Συνδυάζοντας τα υποσυστήματα του παραλήπτη και του αποστολέα μπορούμε να υλοποιήσουμε το συνολικό UART core. Το block διάγραμμα του συνολικού συστήματος φαίνεται στην παρακάτω εικόνα:



- Ο κώδικας του Receiver υποσυστήματος περιέχεται στο αρχείο *uart.v*

### 3.5 Πρόσθετα Αρχεία Κώδικα

#### - Mod M Counter

Ένας mod-m μετρητής, μετράει από το 0 έως το m-1 και πάλι πίσω. Στο project έγινε υλοποίηση ενός mod-m counter με 2 παραμέτρους: M, το οποίο ορίζει το όριο m και το N, το οποίο ορίζει τον αριθμό των bits που απαιτούνται και είναι ίσο με  $\lceil \log_2 M \rceil$ .

- Ο κώδικας του counter περιέχεται στο αρχείο *mod\_m\_counter.v*

#### - FIFO Buffer

Ένας FIFO (First-In-First-Out) buffer είναι μία αποθηκευτική μονάδα μεταξύ δύο υποσυστημάτων. Έχει 2 σήματα ελέγχου, wr και rd, για τις λειτουργίες της εγγραφής και ανάγνωσης αντίστοιχα. Όταν το wr γίνει set, η είσοδος γράφεται στον buffer. Από την άλλη η λειτουργία της ανάγνωσης είναι λίγο παραπλανητική. Η κορυφή του buffer είναι συνέχεια διαθέσιμη και μπορεί να αναγνωστεί οποιαδήποτε στιγμή.

Στην ουσία το rd σήμα αποτελεί ένα σήμα διαγραφής. Όταν γίνει set το πρώτο στοιχείο διαγράφεται και το δεύτερο γίνεται διαθέσιμο προς ανάγνωση.

- Ο κώδικας του FIFO buffer περιέχεται στο αρχείο *fifo.v*

### - Debounce Circuit

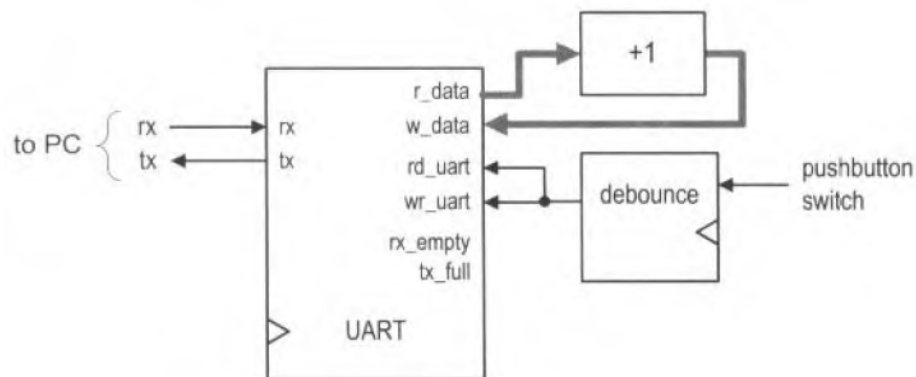
Κύκλωμα για την διαχείριση των bounces των push buttons του αναπτυσσικού.

- Ο κώδικας του FIFO buffer περιέχεται στο αρχείο *debounce.v*

## Κεφάλαιο 4<sup>ο</sup> – Verification Circuits

### 4.1 UART Verification Circuit

Για την επιβεβαίωση της ορθής λειτουργίας του κυκλώματος UART κάνουμε χρήση ενός loop-back κυκλώματος και ένα PC. Το block διάγραμμα του κυκλώματος είναι το εξής:



Η σειριακή θύρα του board συνδέεται με τη σειριακή θύρα του PC. Όταν στέλνουμε ένα χαρακτήρα από το PC, τα ληφθέντα δεδομένα αποθηκεύονται στον 4-word FIFO buffer του receiver. Όταν ληφθεί (μέσω του r\_data port) τα δεδομένα αυξάνονται κατά 1 και αποστέλλονται πίσω στον transmitter (μέσω του w\_data port). Το debounced button παράγει ένα απλό clock-cycle tick όταν πιέζεται και συνδέεται στα σήματα rd\_uart και wr\_uart. Όταν παράγεται το tick, αφαιρείτε μία λέξη από τον FIFO buffer του receiver και γράφει την αυξημένη λέξη στον FIFO buffer του transmitter για να μεταδοθεί.

Για παράδειγμα αρχικά γράφουμε τους χαρακτήρες HAL στο hyperterminal του PC και οι 3 λέξεις αποθηκεύονται στο FIFO buffer του UART receiver. Εάν πατήσουμε το button 3 φορές οι χαρακτήρες IBM θα εμφανιστούν στο hyperterminal. Επίσης τα δεδομένα (8bit) που λαμβάνονται από το receive module (rec\_data) οδηγούν τα 8 LEDs.

### Windows HyperTerminal Options:

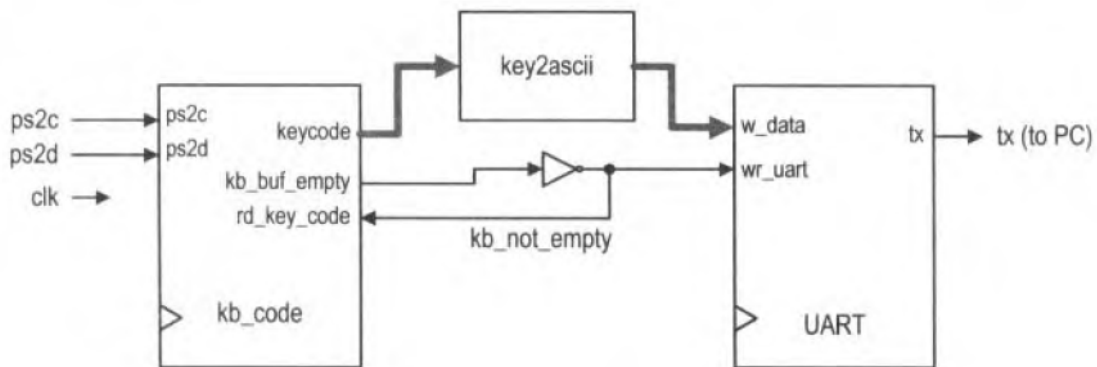
Από την μεριά του PC γίνεται χρήση του προγράμματος HyperTerminal ως ένα virtual terminal για την επικοινωνία με το Spartan Board. Όπως αναφέραμε και παραπάνω πρέπει να υπάρχει συμφωνία σε κάποιες παραμέτρους για να είναι δυνατή η επικοινωνία. Συνεπώς ακολουθούμε τα εξής βήματα:

1. Ανοίγουμε το HyperTerminal
2. Δίνουμε ένα νέο όνομα για την σύνδεση (πχ FPGA\_board)
3. Από το πεδίο Connection επιλέγουμε το serial port (συνήθως COM1).
4. Έπειτα στις ρυθμίσεις δίνουμε τις εξής τιμές:
  - Bits per second: 19200
  - Data bits: 8
  - Parity: None
  - Stop bits: 1
  - Flow control: None
5. Έπειτα πάμε στις ρυθμίσεις της σύνδεσης στο ASCII setup και επιλέγουμε την επιλογή echo ASCII Characters.

- Ο κώδικας του verification κυκλώματος περιέχεται στο αρχείο *uart\_test.v*

### 4.2 PS2 Verification Circuit

Προκειμένου να γίνει έλεγχος της PS2 διασύνδεσης σχεδιάστηκε ένα απλό σειριακό interface και ένα κύκλωμα αποκωδικοποίησης. Το block διάγραμμα του κυκλώματος ελέγχου είναι το εξής:



Το κύκλωμα μετατρέπει το make code του πλήκτρου στην αντίστοιχη ASCII αναπαράσταση και στη συνέχεια στέλνει τον ASCII κωδικό στη UART. Οι ανάλογοι χαρακτήρες ή ψηφία μπορούν να εμφανιστούν στο HyperTerminal ενός PC όπως έγινε και στο παραπάνω παράδειγμα.

- Ο κώδικας του verification κυκλώματος περιέχεται στο αρχείο *kb\_test.v*
- Ο κώδικας του key2ascii κυκλώματος περιέχεται στο αρχείο *key2ascii.v*

## **Βιβλιογραφία – Πηγές**

### **Books:**

- *Digital Design, Third Edition*, by M. Morris Mano
- *Σχεδίαση Ψηφιακών Συστημάτων* by S. Brown, Z. Vranesic
- *Digital Systems Design and Prototyping* by Z. Salcic, A. Smailagic
- *Digital Circuit Analysis and Design with Simulink Modeling* by Steven T. Karris
- *Verilog Tutorial* by D. K. Tala
- *FPGA Prototyping* by Pong P. Chu

### **Datasheets - Manuals:**

- *Spartan 3AN FPGA Starter Kit User Guide*
- *Spartan 3AN FPGA Family Data Sheet*
- *ISE Design Suite Manual*

### **Websites:**

- *Wikipedia*
- *www.xilinx.com*
- *www.fpga4fun.com*
- *www.computer-engineering.org*