



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΗΠΕΙΡΟΥ

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.**

Embedded processor Nios II σε FPGA
της Altera

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



Εισηγητής: Μπούζας Αθανάσιος

Επιβλέπων: Κολιοπάνος Χρήστος

1. ΕΙΣΑΓΩΓΗ

- 1.1 ΣΤΟΧΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ
- 1.2 ΤΙ ΕΙΝΑΙ ΤΟ FPGA
- 1.3 ΕΡΓΑΛΕΙΑ ΠΟΥ ΘΑ ΧΡΗΣΙΜΟΠΟΙΗΣΟΥΜΕ
- 1.4 ΣΧΕΔΙΑΣΜΟΣ ΕΝΟΣ NIOS II ΣΥΣΤΗΜΑΤΟΣ
- 1.5 ΤΟ QUARTUS ΤΗΣ ALTERA
- 1.6 ALTERA MONITOR PROGRAM
- 1.7 ΤΙ ΕΙΝΑΙ ΤΟ SOPC ΤΗΣ ALTERA
- 1.8 ΚΑΘΟΡΙΣΜΟΣ ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ NIOS II ΣΥΣΤΗΜΑΤΟΣ

2. ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

- 2.1 ΞΕΚΙΝΩΝΤΑΣ ΤΟ QUARTUS II ΤΗΣ ALTERA
- 2.2 ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΟΣ ΜΕ ΤΟ SOPC BUILDER
- 2.3 ΕΝΤΑΞΗ ΤΟΥ NIOS II ΣΥΣΤΗΜΑΤΟΣ ΣΤΟ QUARTUS II
- 2.4 ΕΙΣΑΓΩΓΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΕ ΤΟ ALTERA MONITOR PROGRAM

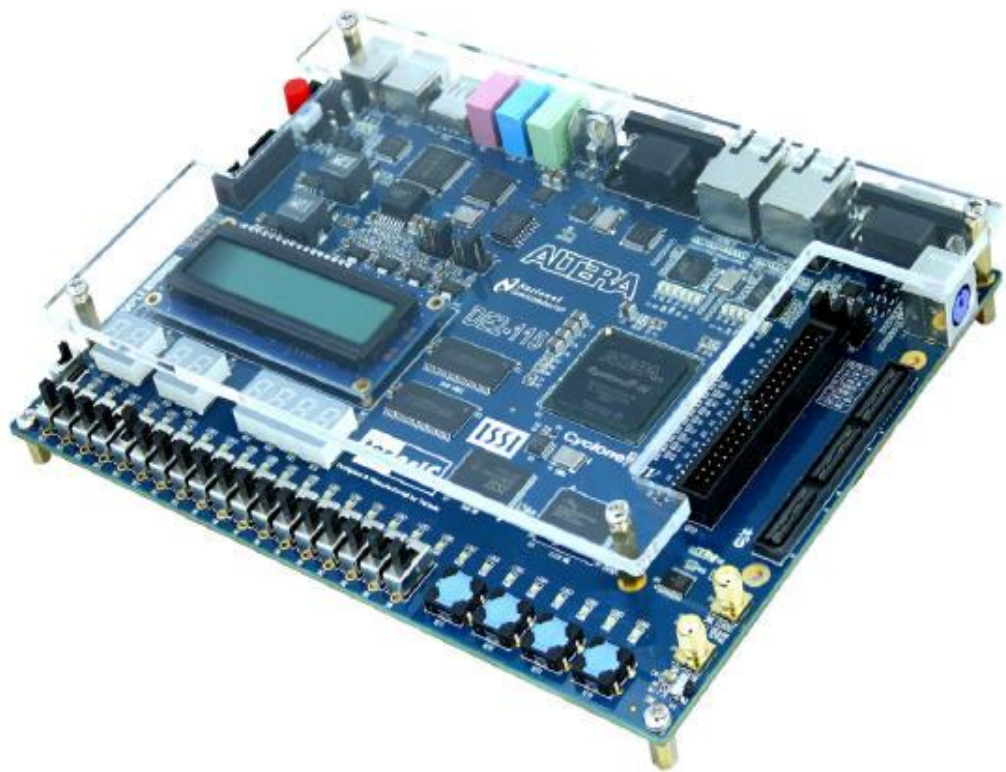
ΔΗΛΩΣΗ ΠΝΕΥΜΑΤΙΚΗΣ ΙΔΙΟΚΤΗΣΙΑΣ

Η παρούσα εργασία αποτελεί προϊόν αποκλειστικά δικής μου προσπάθειας. Όλες οι πηγές που χρησιμοποιήθηκαν περιλαμβάνονται στη βιβλιογραφία και γίνεται ρητή αναφορά σε αυτές μέσα στο κείμενο όπου έχουν χρησιμοποιηθεί.

ΥΠΟΓΡΑΦΗ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ



1.1 Στόχος πτυχιακής εργασίας

Στόχος αυτής της πτυχιακής εργασίας είναι η δημιουργία και η εφαρμογή ενός Nios II συστήματος στην πλακέτα DE2-115 της ALTERA με σκοπό την εμφάνιση μιας εικόνας σε μια LCD οθόνη.

Για την πραγματοποίηση της εργασίας αυτής είναι απαραίτητη η χρήση υλικού (hardware) και λογισμικού (software). Το hardware που θα χρησιμοποιήσουμε είναι η πλακέτα DE2-115 της ALTERA αλλά και μια οθόνη LCD την οποία θα συνδέσουμε με ένα VGA καλώδιο με την πλακέτα έτσι ώστε να μπορέσει να εμφανιστεί η εικόνα που θα έχουμε εισάγει. Η πλακέτα αυτή είναι ένα FPGA το οποίο αναλύετε παρακάτω. Το software που θα χρειαστούμε παρέχεται από την ALTERA μαζί με την πλακέτα. Πρόκειται για το Quartus II 11.0sp1, το οποίο συμπεριλαμβάνει σαν υποπρόγραμμα το SOPC Builder αλλά και το Altera Monitor Program.

1.2 Τι είναι το FPGA

Το FPGA ή Field Programmable Gate Array είναι ένας τύπος προγραμματιζόμενου κυκλώματος γενικής χρήσης το οποίο διαθέτει πολύ μεγάλο αριθμό τυποποιημένων πυλών και άλλων ψηφιακών λειτουργιών όπως απαριθμητές, καταχωρητές μνήμης, γεννήτριες PLL κ.α. Σε ορισμένα από αυτά ενσωματώνονται και αναλογικές λειτουργίες. Κατά τον προγραμματισμό του FPGA, ο οποίος γίνεται πάντοτε ενώ αυτό είναι τοποθετημένο στο τυπωμένο κύκλωμα, ενεργοποιούνται οι επιθυμητές λειτουργίες και διασυνδέονται μεταξύ τους έτσι ώστε το FPGA να συμπεριφέρεται ως ολοκληρωμένο κύκλωμα με συγκεκριμένη λειτουργία.

Ο κώδικας με τον οποίο προγραμματίζεται το FPGA γράφεται σε γλώσσες περιγραφής υλικού (VHDL, AHDL, Verilog).

Το FPGA έχει παρόμοιο πεδίο εφαρμογών με άλλα προγραμματιζόμενα ολοκληρωμένα ψηφιακά κυκλώματα όπως τα PLD και τα ASIC. Όμως τα ιδιαίτερα χαρακτηριστικά του FPGA είναι τα εξής :

Το FPGA χάνει τον προγραμματισμό του κάθε φορά που διακόπτεται η τάση τροφοδοσίας του. Επομένως απαιτεί εξωτερικό μικροεπεξεργαστή ή μνήμη με μόνιμη συγκράτηση δεδομένων (non-volatile memory) από τα οποία θα προγραμματίζεται κάθε φορά που επανέρχεται η τάση τροφοδοσίας.

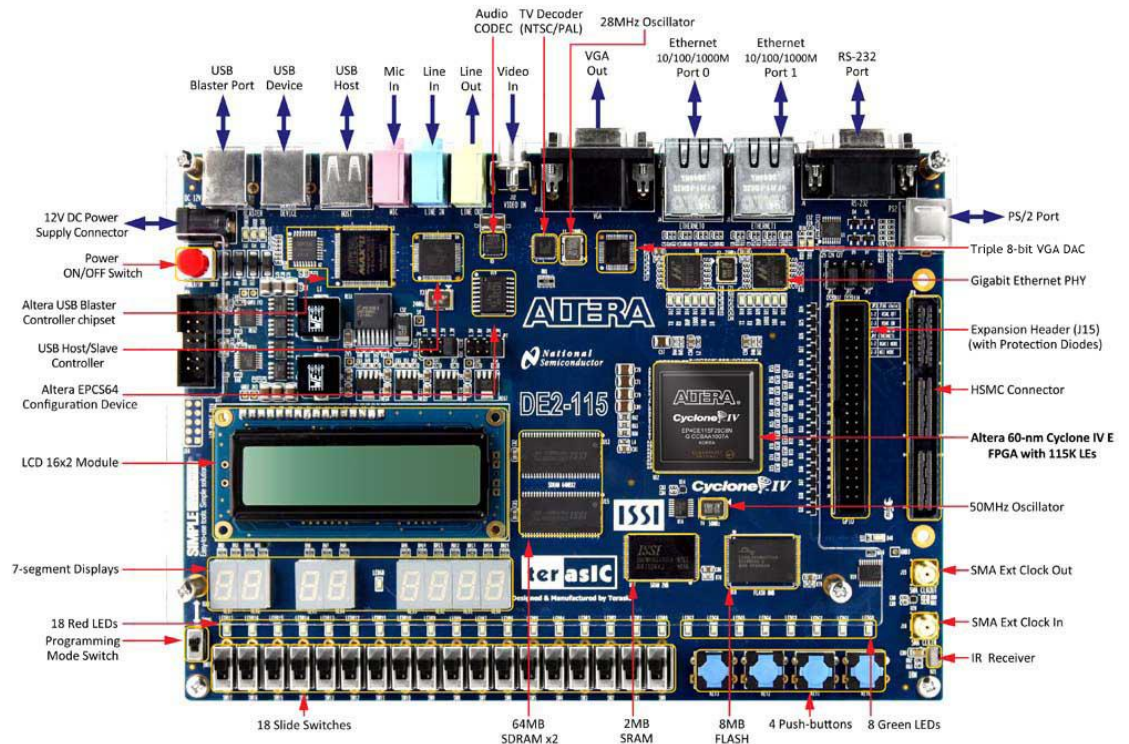
Ο προγραμματισμός του FPGA μπορεί να αλλάζει κάθε φορά που τροποποιείται το λογισμικό του μικροεπεξεργαστή ή τα δεδομένα της μνήμης που το ελέγχει. Δεν υπάρχει όριο στο πόσες φορές μπορεί να επαναπρογραμματιστεί. Η κατανάλωση ισχύος είναι σημαντικά αυξημένη σε σχέση με τα ASIC.

Έτσι το FPGA είναι ιδιαίτερα κατάλληλο εκεί που οι παράμετροι λειτουργίας πρέπει να αλλάζουν συχνά ή σε μικρές ποσότητες παραγωγής, ενώ τα ASIC, λόγω μαζικής παραγωγής, είναι φτηνότερα εκεί που απαιτούνται μεγάλες ποσότητες και η επιθυμητή λειτουργία είναι αυστηρά προκαθορισμένη, χωρίς σφάλματα (το ASIC δεν επαναπρογραμματίζεται).

Βασική δομική μονάδα του FPGA είναι το λογικό μπλοκ, με τη χρήση του οποίου υλοποιούνται οι λογικές συναρτήσεις που εκφράζουν τη λειτουργία ενός ψηφιακού κυκλώματος. Ανάλογα με το μέγεθος του κυκλώματος πολλά λογικά μπλοκ συνδέονται για να υλοποιήσουν το πλήθος των απαραίτητων λογικών συναρτήσεων.

1.3 Εργαλεία που θα χρησιμοποιήσουμε

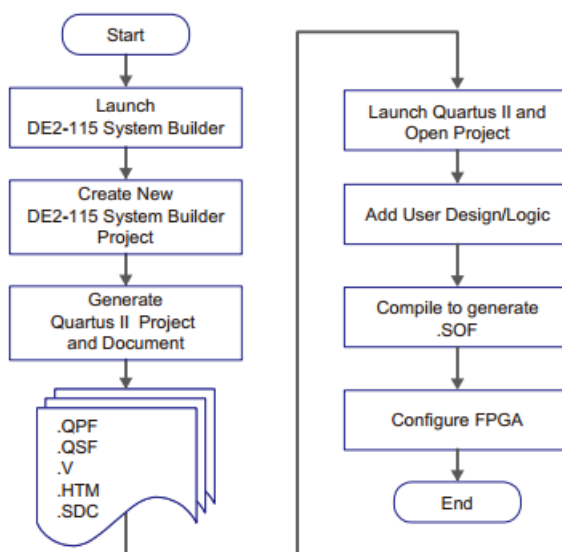
- Quartus II 11.0sp1
- SOPC Builder (εργαλείο του Quartus II)
- Altera Monitor Program
- Πλακέτα DE2-115 της ALTERA
- Μια οθόνη LCD



Εικόνα 1. Η DE2-115 της ALTERA

1.4 Σχεδιασμός ενός Nios II συστήματος

Ο σχεδιασμός ενός Nios II συστήματος, αποτελείται από τρεις τύπους σχεδίασης: τον σχεδιασμό υλικού, τον σχεδιασμό λογισμικού και τον σχεδιασμό του συστήματος ο οποίος συμπεριλαμβάνει υλικό αλλά και λογισμικό. Για ποιά απλά Nios II συστήματα ένας άνθρωπος θα μπορούσε να υλοποιήσει και τους τρεις τύπους σχεδίασης. Για ποια πολύπλοκα συστήματα χρειάζονται ξεχωριστές ομάδες ανθρώπων για τον σχεδιασμό υλικού και λογισμικού. Σε αυτές τις περιπτώσεις ο σχεδιασμός του συστήματος θα χρειαστεί να υλοποιηθεί και από τις δυο ομάδες από τις οποίες θα παίρνει πληροφορίες. Κατά τον σχεδιασμό είναι πολύ σημαντικό οι δύο ομάδες να γνωρίζουν όλες τις πληροφορίες στα σημεία όπου διασταυρώνονται το υλικό και το λογισμικό. Στο σχήμα 1 βλέπουμε το μοντέλο σχεδιασμού ενός Nios II συστήματος. Στην εργασία αυτή επικεντρωνόμαστε κυρίως στο σχεδιασμό υλικού.



Σχήμα 1. Το γενικό μοντέλο σχεδιασμού ενός συστήματος Nios II

1.5 Το Quartus της ALTERA

Το Quartus είναι ένα πρόγραμμα σχεδιασμού λογισμικού της ALTERA. Τα κύρια χαρακτηριστικά του είναι τα εξής:

1. Η εφαρμογή της VHDL και Verilog γλώσσας για την περιγραφή του υλικού.
2. Οπτική περιγραφή των λογικών κυκλωμάτων.

3. Προσομοίωση της κυματομορφής του διανύσματος.

Το Quartus II, το οποίο θα χρησιμοποιήσουμε και στην συγκεκριμένη εργασία, είναι ένα πρόγραμμα που παρέχεται από την ALTERA για την ανάλυση και την σύνθεση HDL σχεδίων, το οποίο επιτρέπει στον χρήστη να συντάξει και να μεταγλωττίσει ένα σχέδιο, να μελετήσει διαγράμματα RTL, να προσομοιώσει το σχέδιο και να παρατηρήσει τυχόν αντιδράσεις σε αυτό αλλά και αλλαγές οι οποίες θα βοηθήσουν στην βελτίωση του. Τέλος σου δίνει την δυνατότητα να ρυθμίσουμε τις παραμέτρους για την σύνδεση του FPGA με την συσκευή προορισμού, στην προκειμένη περίπτωση την οθόνη LCD. Η τελευταία έκδοση του λογισμικού είναι η 13sp1. Η έκδοση αυτή υποστηρίζει τις εξής συσκευές : Cyclone III, Cyclone IV, MAX II και MAX V σε όλη την γκάμα τους καθώς και μερικές από τις Arria II και Cyclone V.

1.6 Altera Monitor Program

Το Altera Monitor Program είναι ένα πρόγραμμα το οποίο παρέχεται από την ALTERA με το University Program Installer και επιτρέπει στο χρήστη πολύ εύκολα να μεταγλωττίσει και να εντοπίσει σφάλματα σε γλώσσα Assembly αλλά και σε προγράμματα γραμμένα σε γλώσσα C. Εμφανίζει την κατάσταση του Nios II επεξεργαστή κατά την διάρκεια της εκτέλεσης του προγράμματος, όπως για παράδειγμα την κατάσταση της μνήμης του επεξεργαστή. Επίσης επιτρέπει πολύ εύκολα να ορίσει κάποιος τα σημεία διακοπής κατά την διάρκεια εκτέλεσης του προγράμματος.

Το πρόγραμμα αυτό δε υποστηρίζει συστήματα που έχουν ανατηχθεί χρησιμοποιώντας το νέο εργαλείο του Quartus II, Qsys. Οπότε είναι απαραίτητο να χρησιμοποιηθεί το SOPC Builder αν θέλει ο χρήστης να χρησιμοποιήσει και το Altera Monitor Program.

1.7 Τι είναι το SOPC της ALTERA

Το SOPC (SOPC Builder) είναι ένα εργαλείο που χρησιμοποιείται σε συνδυασμό με το λογισμικό Quartus II. Επιτρέπει στο χρήστη να δημιουργήσει εύκολα ένα σύστημα βασισμένο στον επεξεργαστή Nios II, επιλέγοντας απλά τις επιθυμητές λειτουργικές μονάδες και την επιλογή των παραμέτρων τους.

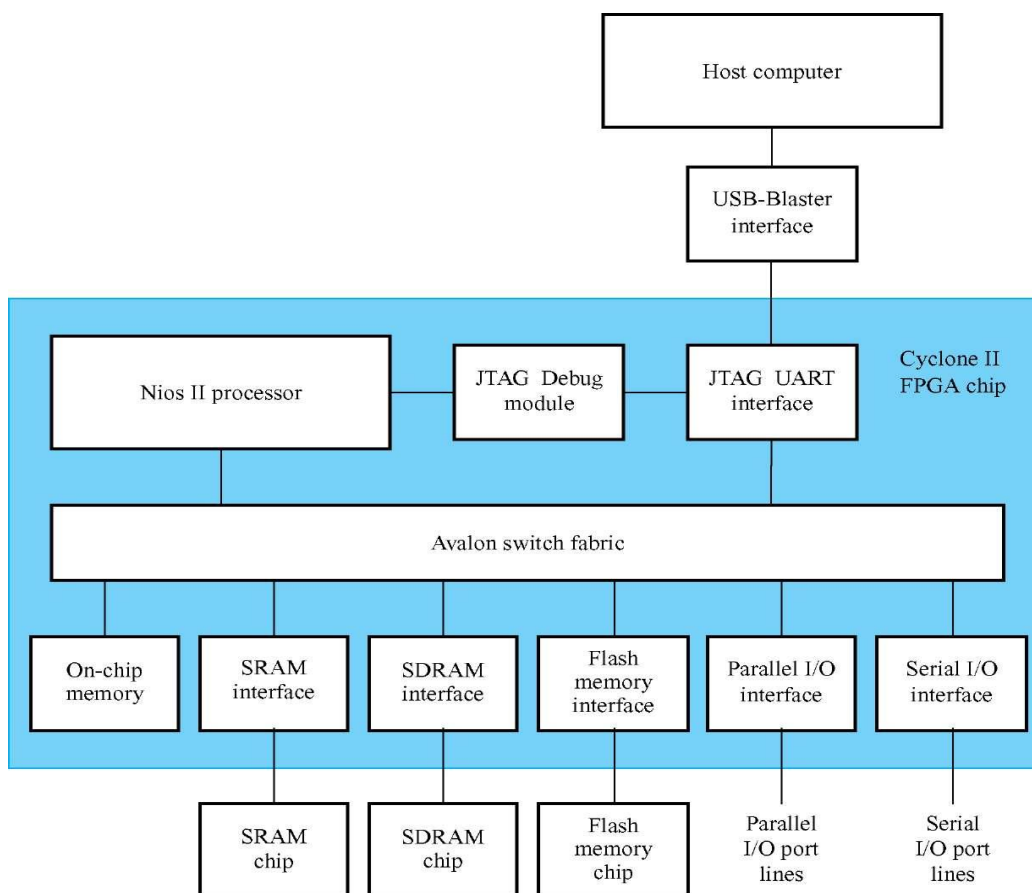
1.8 Καθορισμός και δημιουργία ενός Nios II συστήματος

Αφού κάνουμε μια ανάλυση των απαιτήσεων του υλικού του συστήματος, με την βοήθεια του SOPC Builder θα καθορίσουμε τον Nios II επεξεργαστή, τις μνήμες καθώς και τα υπόλοιπα στοιχεία που απαιτεί το σύστημα. Το SOPC Builder αυτόματα δημιουργεί την διασύνδεση μεταξύ των λογικών στοιχείων που χρειάζεται για να ενσωματωθούν στο σύστημα.

Μπορούμε να επιλέξουμε από μία λίστα με τυποποιημένους επεξεργαστές και άλλα στοιχεία που παρέχονται από το πρόγραμμα Nios II EDS. Επίσης μπορούμε να προσθέσουμε το δικό μας προσαρμοσμένο υλικό για να επιταχύνουμε την απόδοση του συστήματος καθώς και να επιταχύνουμε τον επεξεργαστή ή ακόμα και προσθέτοντας κάποια στοιχεία να τον αποφορτίσουμε.

Το SOPC Builder θα δημιουργήσει κάποια αρχεία τα οποία θα περιέχουν όλες τις απαραίτητες πληροφορίες σχετικά με το σχεδιασμό του υλικού και τη διασύνδεση του.

Αφού δημιουργήσουμε το Nios II σύστημα με το SOPC Builder, το ενσωματώνει στο project του προγράμματος Quartus II. Στη συνέχεια με την βοήθεια του προγράμματος αυτού μπορούμε να εκτελέσουμε όλες τις εργασίες που απαιτούνται για να δημιουργήσουμε το τελικό σχέδιο το οποίο θα προσομοιώσουμε στο FPGA. Στο σχήμα 2 μπορούμε να δούμε ένα παράδειγμα συστήματος που μπορεί να εφαρμοστεί σε ένα FPGA.



Σχήμα 2. Ένα Nios II σύστημα

ΚΕΦΑΛΑΙΟ 2

ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

2.1 ΞΕΚΚΙΝΩΝΤΑΣ ΤΟ QUARTUS II ΤΗΣ ALTERA

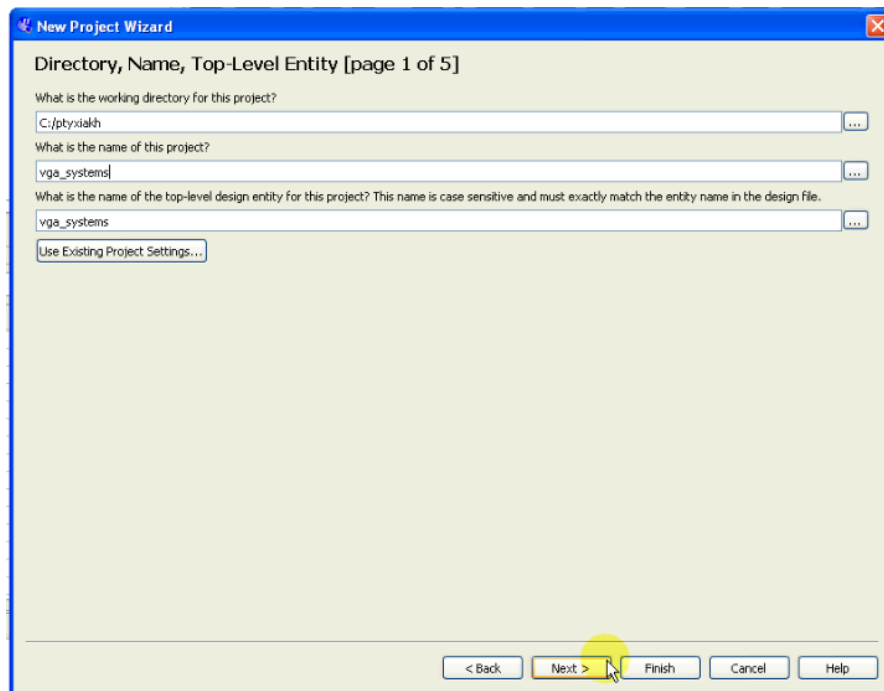
Για να ξεκινήσουμε ανοίγουμε το πρόγραμμα Quartus II. Με αυτό το πρόγραμμα θα δημιουργήσουμε το project ακολουθώντας τα παρακάτω βήματα :

1. Κάνουμε κλικ στο **Create a new project** στον οδηγό δημιουργίας του project.



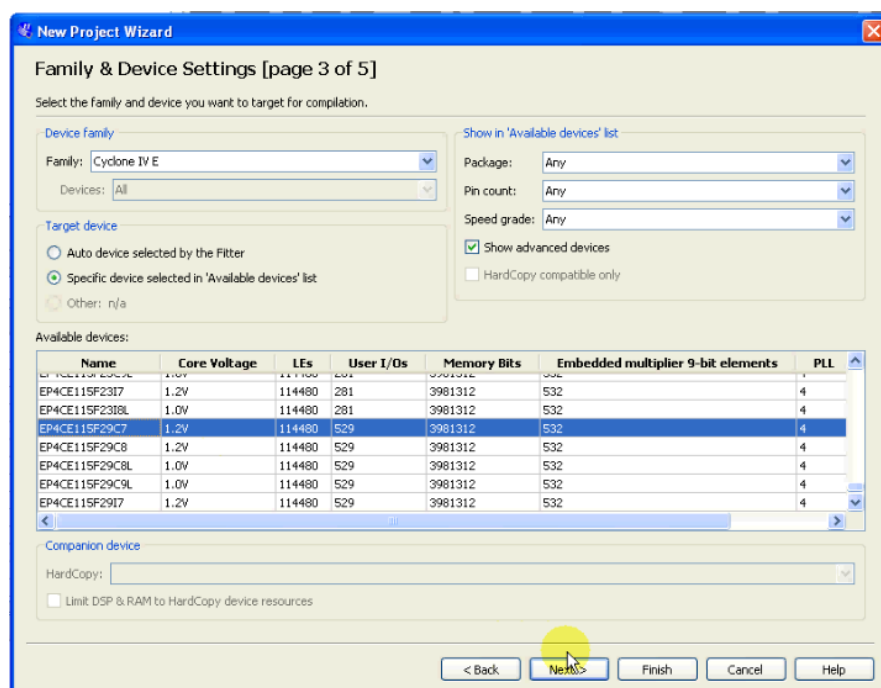
Εικόνα 2. Ο οδηγός δημιουργίας Project

2. Πατάμε **Next**. Στη σελίδα που εμφανίζετε επιλέγουμε που θα αποθηκευτούν τα αρχεία που θα δημιουργηθούν από το Quartus II. Πρέπει να προσέξουμε να μην υπάρχουν κενά στο μονοπάτι όπου το αποθηκεύουμε. Σε αυτή τη περίπτωση το αποθηκεύουμε στο **C** στον φάκελο **ptyxiakh**. Στη συνέχεια δίνουμε το όνομα **vga_systems** στο project και πατάμε **Next**.



Εικόνα 3. Τα στοιχεία του project

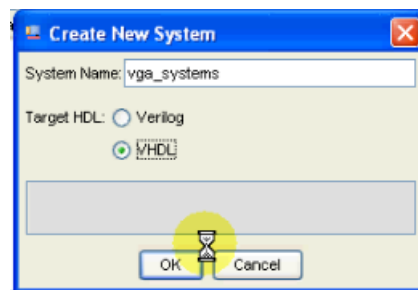
3. Πατάμε και πάλι **Next** καθώς δεν χρειάζεται να κάνουμε κάποια αλλαγή. Στη σελίδα που εμφανίζεται θα επιλέξουμε το μοντέλο της πλακέτας που χρησιμοποιούμε. Στο **Device family** θα επιλέξουμε τον **Cyclone IV E**. Στο **Target device** επιλέγουμε **Specific device selected in 'Available devices' list**. Στη συνέχεια επιλέγουμε το **EP4CE115F29C7** από την λίστα και πατάμε **Next**. Πατάμε και πάλι **Next** αφού δεν χρειάζεται να αλλάξουμε κάτι και μετά **Finish**. Έτσι έχουμε ολοκληρώσει τον οδηγό για την δημιουργία του Project.



Εικόνα 4. Family & Device Settings

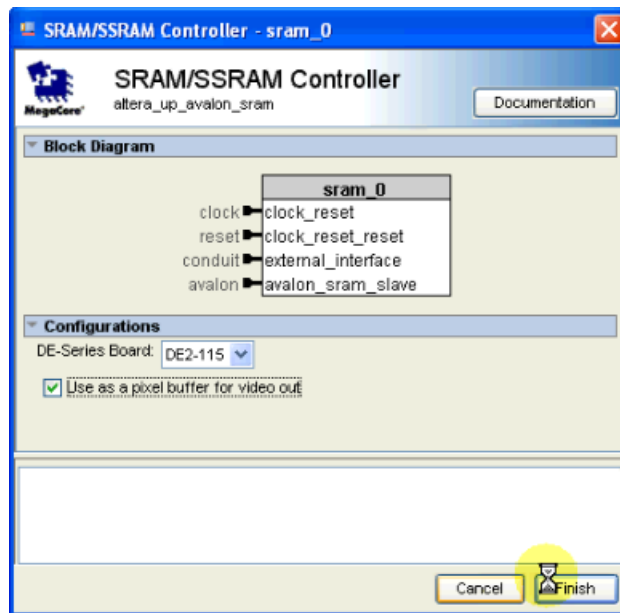
2.2 ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΤΗΜΑΤΟΣ ΜΕ ΤΟ SOPC BUILDER

Για να ανοίξουμε το SOPC Builder επιλέγουμε **tools > SOPC Builder**. Δίνουμε το όνομα **vga_systems** στο σύστημα και επιλέγουμε **VHDL** στον οποίο η ενότητα συστημάτων θα διευκρινιστεί, όπως φαίνεται στην εικόνα 5, και πατάμε **ok**. Το SOPC Builder έχει έτοιμη την πρώτη διεπαφή η οποία είναι το ρολόι του συστήματος που ελέγχει την ταχύτητα του επεξεργαστή. Σε αυτό το παράδειγμα θα χρησιμοποιήσουμε το ρολόι **50.0 Mhz** που παρέχετε στην πλακέτα **DE2-115**. Επίσης υπάρχει η δυνατότητα να αλλάξουμε το όνομα και την ταχύτητα του ρολογιού. Εάν όχι, προσθέτουμε ένα ρολόι που ονομάζεται **clk** με την πηγή ως εξωτερική (**external**) και την συχνότητα καθορισμένη στα **50.0 Mhz**. Στη συνέχεια θα τοποθετήσουμε τα εξαρτήματα που χρειάζονται για να δημιουργηθεί το σύστημα ακολουθώντας τα παρακάτω βήματα :



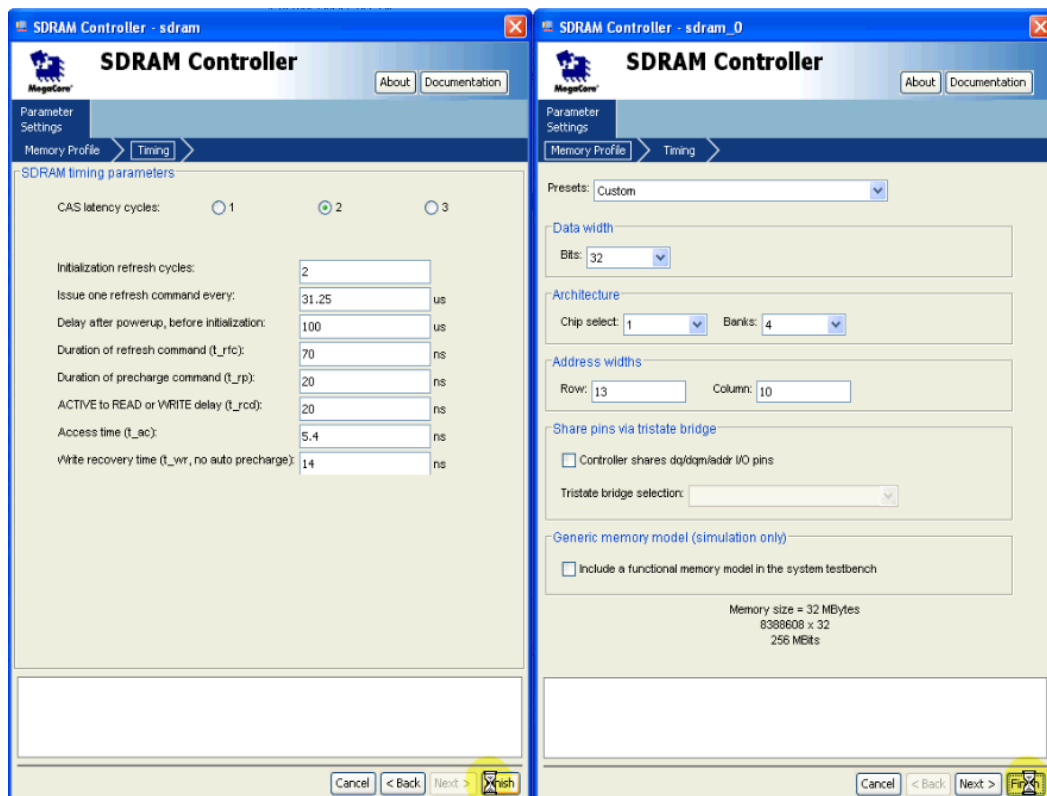
Εικόνα 5. Δημιουργία συστήματος στο SOPC Builder

1. Στη βιβλιοθήκη εξαρτημάτων επιλέγουμε το **Nios II processor** όπου θα είναι ο επεξεργαστής που θα χρησιμοποιήσουμε και πατάμε **add**. Θα εμφανιστεί ένα παράθυρο για τις ρυθμίσεις του επεξεργαστή. Σε αυτό επιλέγουμε τον **Nios II/E** όπου είναι η πιο απλή έκδοση του Nios II και πατάμε **finish** καθώς τις υπόλοιπες ρυθμίσεις θα τις κάνουμε αργότερα. Πατώντας δεξί κλικ στο όνομα του επεξεργαστή και μετά **rename** αλλάζουμε το όνομα σε **cpu**.
2. Στη συνέχεια θα επιλέξουμε από τη βιβλιοθήκη τον **SRAM/SSRAM Controller** και πατάμε **add**. Στο παράθυρο των ρυθμίσεων που εμφανίζεται επιλέγουμε στο μοντέλο της πλακέτας την **DE2-115** και επιλέγουμε το **Use as a pixel buffer for video out** καθώς αυτή θα είναι η μνήμη την οποία θα χρησιμοποιεί το σύστημα σαν buffer. Μετά πατάμε **finish** και το μετονομάζουμε σε **sram**.



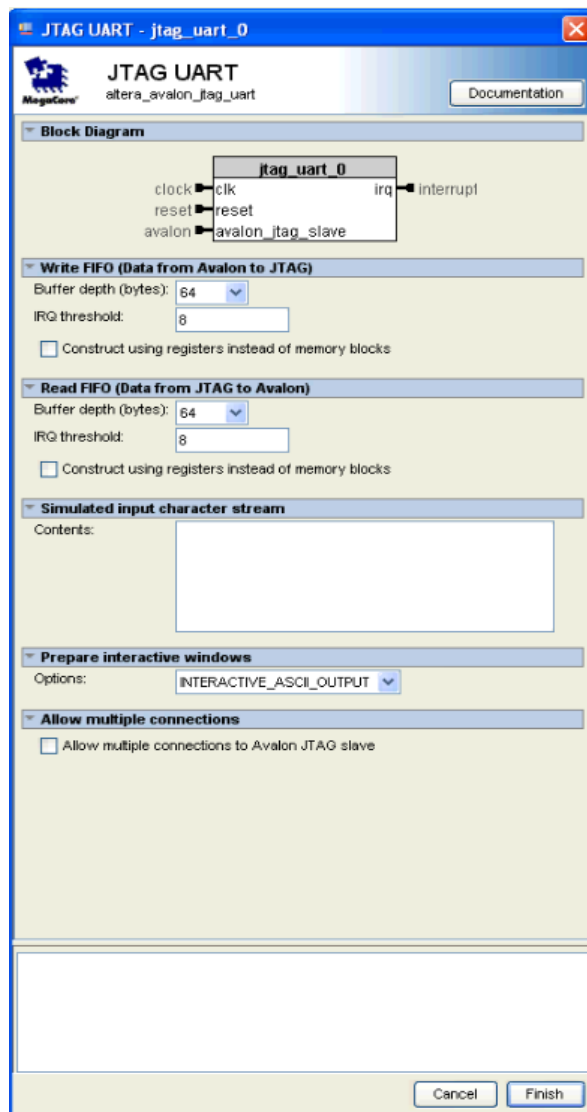
Εικόνα 6. SRAM/SSRAM Controller Editor

3. Στη βιβλιοθήκη επιλέγουμε το **SDRAM Controller** ,όπου θα είναι η μνήμη του συστήματος, και πατάμε **add**. Στο παράθυρο που εμφανίζεται κάνουμε τις εξής ρυθμίσεις. Στη καρτέλα **Memory profile** στο **presets** επιλέγουμε **custom**. Στο **Address widths** βάζουμε στο **Row** 13 και στο **Column** 10. Στην καρτέλα **Timing** επιλέγουμε στο **CAS latency cycles** to 2, **Issue one refresh command every** βάζουμε 31.25, και στο **Access time** 5.4. Τις υπόλοιπες ρυθμίσεις τις αφήνουμε ως έχουν. Μετά πατάμε **finish** και το μετονομάζουμε σε **sdrām**.



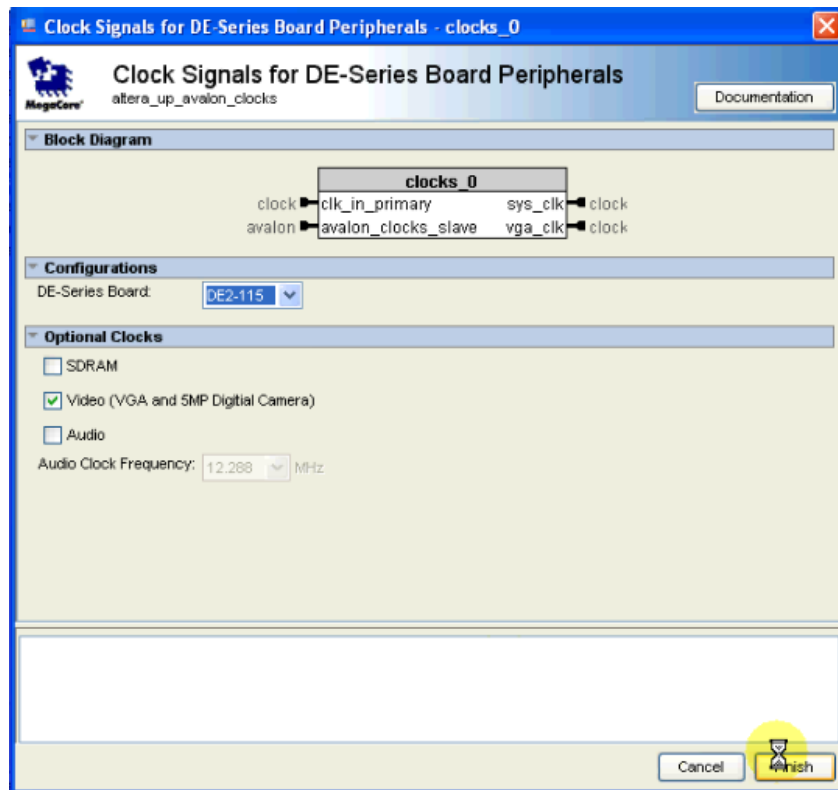
Εικόνα 7. SDRAM Controller Editor

4. Για να μπορούμε να συνδέουμε την πλακέτα με τον Η/Υ προσθέτουμε την διεπαφή **JTAG UART**. Η διεπαφή αυτή είναι υπεύθυνη για την επικοινωνία μεταξύ του FPGA και του Η/Υ. Οπότε το επιλέγουμε και πατάμε add. Δεν χρειάζεται να κάνουμε κάποια αλλαγή στις ρυθμίσεις στο παράθυρο που εμφανίζεται. Πατάμε **finish** και το μετονομάζουμε σε **jtag_uart**.



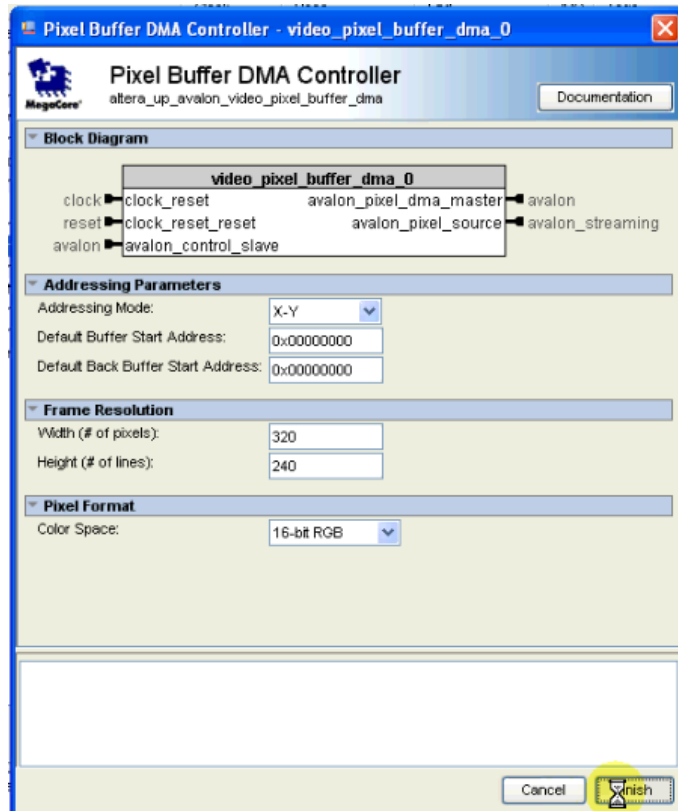
Εικόνα 8. JTAG/UART Editor

5. Η επόμενη διεπαφή που θα προσθέσουμε είναι το **Clock Signals for DE-Series Board Peripherals**. Η διεπαφή αυτή μπορεί να δώσει διαφορετικές συχνότητες από αυτή του κυρίως ρολογιού. Στο συγκεκριμένο παράδειγμα χρειαζόμαστε συχνότητα **25.0 Mhz** για να λειτουργήσει η οθόνη. Στο παράθυρο που εμφανίζεται ξετσεκάρουμε τα **Audio** και **SDRAM** και στο **Configurations** αλλάζουμε το **DE-Series Board** σε **DE2-115** που είναι και η πλακέτα που χρησιμοποιούμε. Πατάμε **finish** και το μετονομάζουμε σε **external_clocks**.



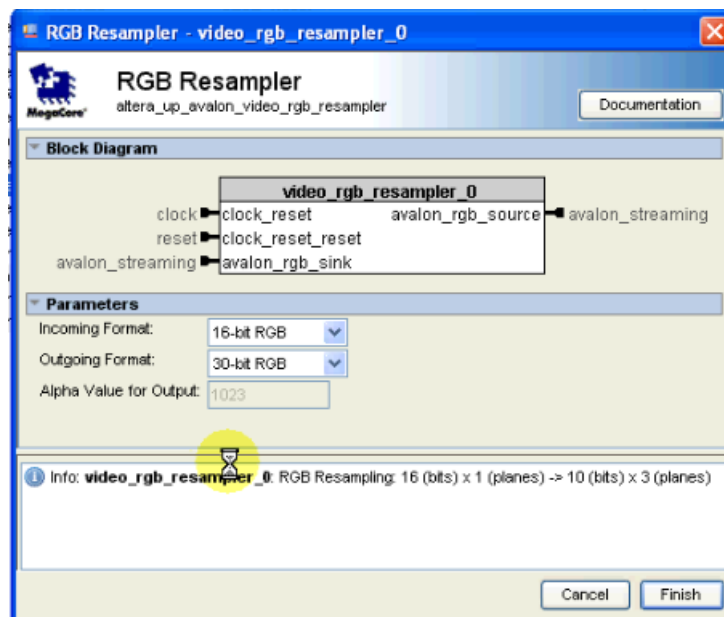
Εικόνα 9. External Clocks Editor

6. Επιλέγουμε από την βιβλιοθήκη τη διεπαφή **Pixel Buffer DMA Controller** και πατάμε **add**. Η διεπαφή αυτή παράγει τα δεδομένα του Video και τα στέλνει στον VGA Controller. Ενώ δεν περιέχει μνήμη σου παρέχει την δυνατότητα να την συνδέσεις με εξωτερική μνήμη που θα παίζει τον ρόλο του buffer. Στις ρυθμίσεις θα αλλάξουμε στο **Frame Resolution** το **Width** σε 320 και το **Height** σε 240. Αυτή θα είναι και η ανάλυση της εικόνας. Πατάμε **finish** και μετονομάζουμε σε **vga_pixel_buffer**.



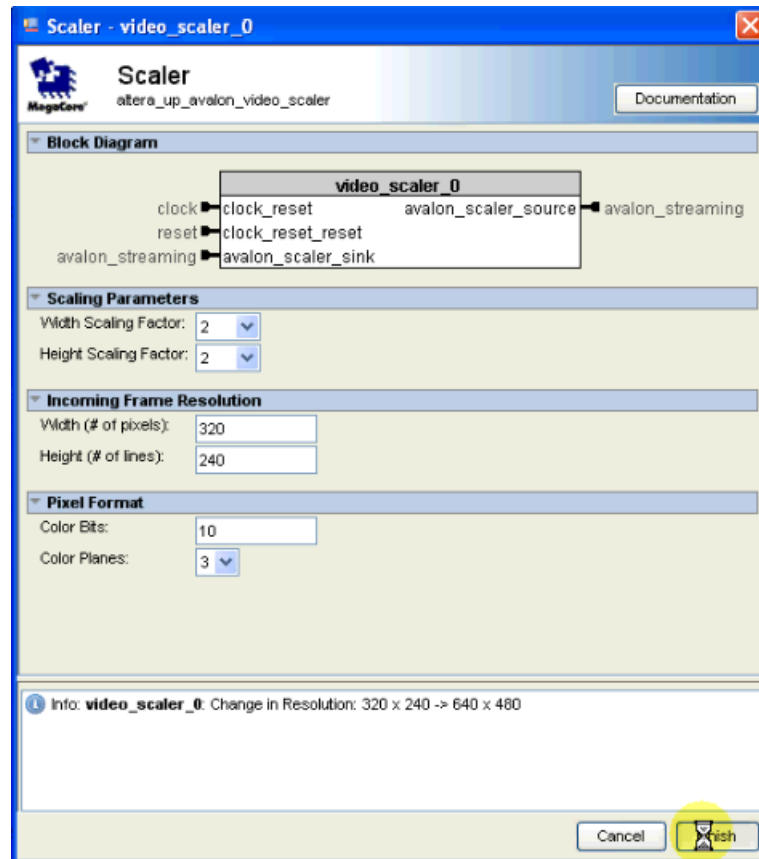
Εικόνα 10. VGA_Pixel_Buffer Editor

7. Στη συνέχεια θα προσθέσουμε την διεπαφή **RGB Resampler**. Η διεπαφή αυτή μετατρέπει το σήμα από 16-bit που είναι το σήμα εισόδου σε 30-bit σήμα εξόδου όπου είναι απαραίτητο για την λειτουργία της οθόνης. Δεν χρειάζονται κάποιες αλλαγές. Πατάμε **finish** και μετονομάζουμε σε **video_rgb_resampler**.



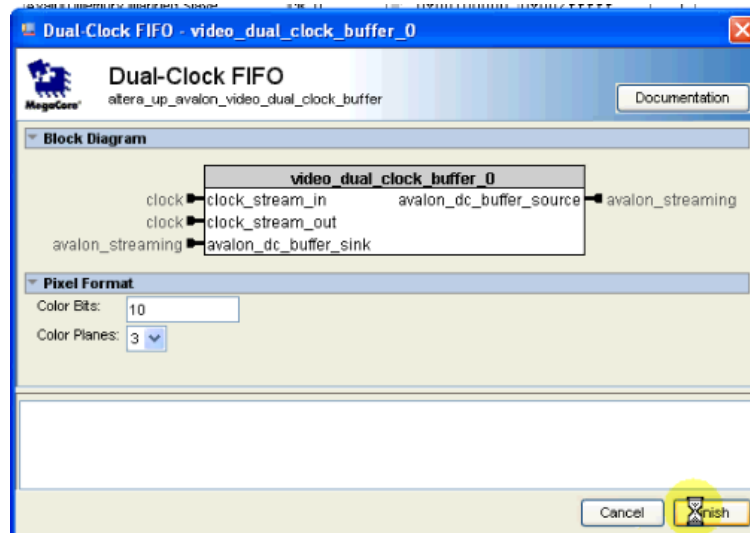
Εικόνα 11. Video_RGB_Resampler Editor

8. Από την βιβλιοθήκη προσθέτουμε το **Video Scaler** πατώντας **add**. Πρόκειται για μια διεπαφή που σου δίνει την δυνατότητα να αλλάξεις την αρχική ανάλυση της εικόνας. Στο παράθυρο των ρυθμίσεων αλλάζουμε στο **Scaling Parameters** τα **Width Scaling Factor** και **Height Scaling Factor** σε 2. Στο **Incoming Frame Resolution** αλλάζουμε τα **Width** και **Height** σε 320 και 240 αντίστοιχα ενώ στο **Pixel Format** αλλάζουμε τα **Color Bits** σε 10. Πατάμε finish και το μετονομάζουμε σε video_scaler.



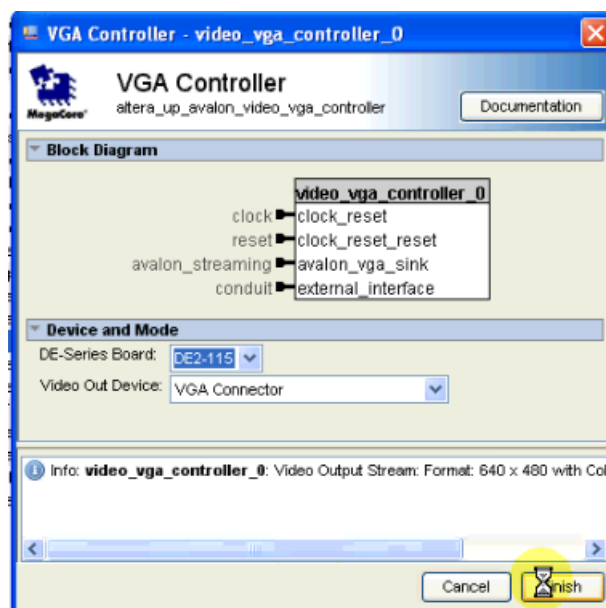
Εικόνα 12. Video_Scaler Editor

9. Η επόμενη διεπαφή που θα προσθέσουμε είναι το **Dual-Clock FIFO** το οποίο έχει ως σκοπό να δίνει την σωστή συχνότητα στον VGA Controller. Αφού πατήσουμε **add**, στο παράθυρο που εμφανίζεται πατάμε **finish** μιας και δεν χρειάζεται να κάνουμε κάποια αλλαγή και το μετονομάζουμε σε **dual_clock_fifo**.



Εικόνα 13. Dual_Clock_FIFO Editor

10. Τέλος προσθέτουμε τον **VGA Controller** από την βιβλιοθήκη. Ο VGA Controller περιγράφει τις διαστάσεις πλάτους και ύψους μιας οθόνης, όπως μια οθόνη H/Y, σε pixel. Ορισμένοι συνδυασμοί πλάτους και ύψους είναι τυποποιημένοι. Υψηλότερη ανάλυση (Display Resolution) σημαίνει ποιο καθαρή εικόνα. Αφού πατήσουμε **add** στο παράθυρο που εμφανίζετε θα αλλάξουμε από το **Device and Mode** το **DE-Series Board** και θα το βάλουμε **DE2-115**. Πατάμε finish και το μετονομάζουμε σε **VGA_Controller**.



Εικόνα 14. VGA_Controller Editor

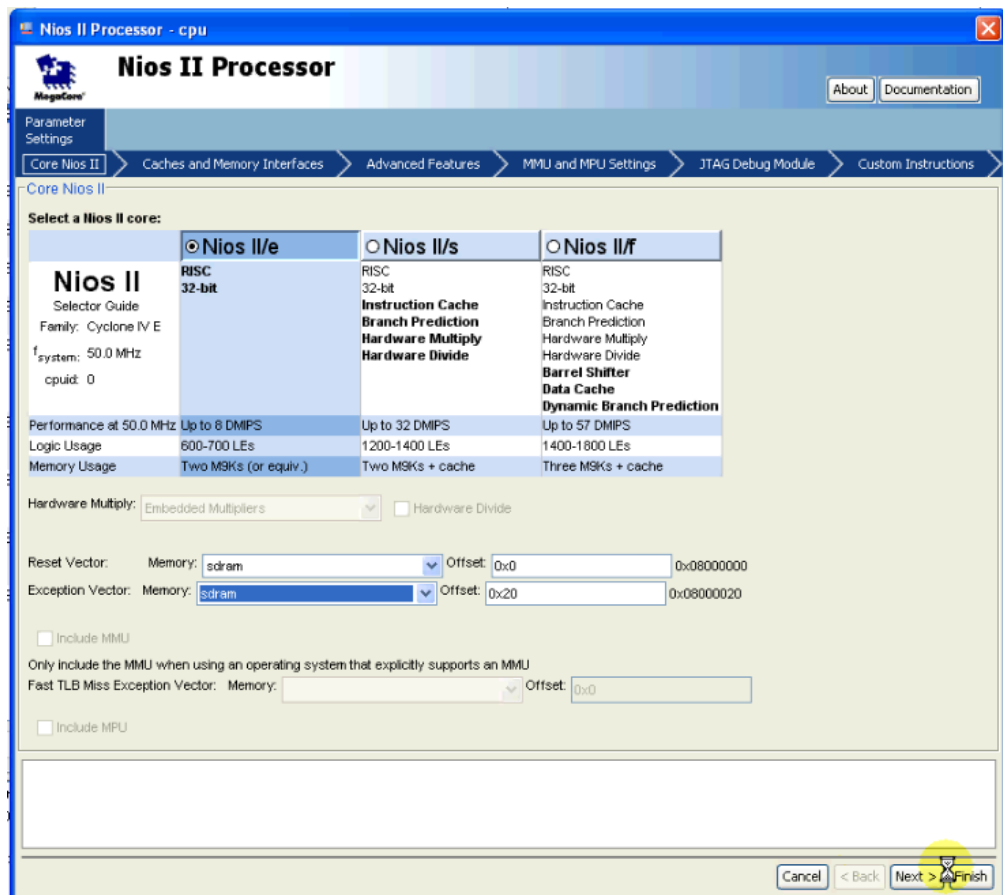
Έχοντας τελειώσει με την προσθήκη των διεπαφών χρειάζεται να κάνουμε τις απαραίτητες συνδέσεις για να λειτουργήσουν. Για να συνδέσουμε μια διεπαφή με μια άλλη πηγαίνουμε πάνω στις εισόδους των διεπαφών και τότε εμφανίζονται όλες οι δυνατές συνδέσεις. Για να ενεργοποιήσουμε κάποια σύνδεση κάνουμε κλικ πάνω στην κουκίδα που βρίσκεται στη συγκεκριμένη σύνδεση.

Οι συνδέσεις που θα κάνουμε είναι οι εξής :

1. Συνδέουμε το **Avalon_pixel_dma_master** από το **vga_pixel_buffer** με το **Avalon_sram_slave** από την **sram**.
2. Από το **vga_pixel_buffer** το **Avalon_pixel_source** με το **Avalon_rgb_sink** από το **video_rgb_resampler**.
3. Το **Avalon_rgb_source** από το **video_rgb_resampler** με το **Avalon_scaler_sink** από το **video_scaler**.
4. Το **Avalon_scaler_source** από το **video_scaler** με το **Avalon_dc_buffer_sink** από το **dual_clock_fifo**.
5. Το **Avalon_dc_buffer_source** από το **dual_clock_fifo** με το **Avalon_vga_sink** από τον **vga_controller**.

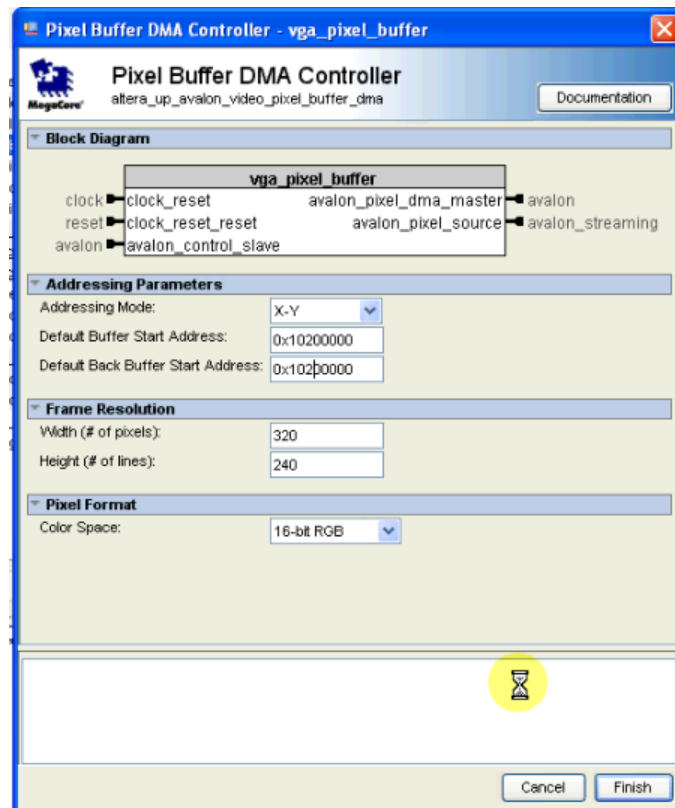
Στη συνέχεια θα ρυθμίσουμε τα ρολόγια του συστήματος. Στο πάνω μέρος της οθόνης στο πλαίσιο **clock settings** υπάρχουν τα διαθέσιμα ρολόγια. Θα μετονομάσουμε το **clk_0** σε **clk_50**, το **external_clocks_sys_clk** σε **sys_clk** και το **external_clocks_vga_clk** σε **vga_clk**. Στην καρτέλα **clock** των διεπαφών επιλέγουμε το **clk_50** για την διεπαφή **external_clocks**, το **vga_clk** για τις διεπαφές **dual_clock_fifo** και **vga_controller**. Για όλες τις υπόλοιπες θα επιλέξουμε το **sys_clk**.

Στη καρτέλα **IRQ** βάζουμε την τιμή **8** στο **jtag_uart**. Πηγαίνουμε στο **System > Assign Base Addresses** και κάνουμε κλικ. Έτσι δίνουμε σε όλες τις διεπαφές καινούρια διεύθυνση. Κάνουμε διπλό κλικ στη **cpu** και ανοίγει το παράθυρο των ρυθμίσεων. Στο **Reset Vector Memory** αλλά και στο **Exception Vector Memory** επιλέγουμε την **sdram** πατάμε **finish**.



Εικόνα 15. CPU Editor

Μετά ανοίγουμε το παράθυρο ρυθμίσεων από το **VGA_Pixel_buffer** και στο **Default Buffer Start Address** αλλά και στο **Default Back Buffer Start Address** βάζουμε την τιμή της διεύθυνσης από την καρτέλα **Base** της **sram**, στην προκειμένη περίπτωση (0x10200000). Μετά πατάμε **finish**.



Εικόνα 16. VGA_Pixel_Buffer Editor

Το σύστημα είναι έτοιμο τώρα όπως φαίνεται και στην εικόνα 17. Πατάμε **Generate** και το αποθηκεύουμε με το όνομα vga_systems. Στη συνέχεια το SOPC Builder δημιουργεί το σύστημα, κάτι που μπορεί να χρειαστεί λίγα λεπτά. Όταν ολοκληρωθεί θα μας βγάλει το μήνυμα “**System generation was successful**”. Ελαχιστοποιούμε το παράθυρο και επιστρέφουμε στο **Quartus II**.

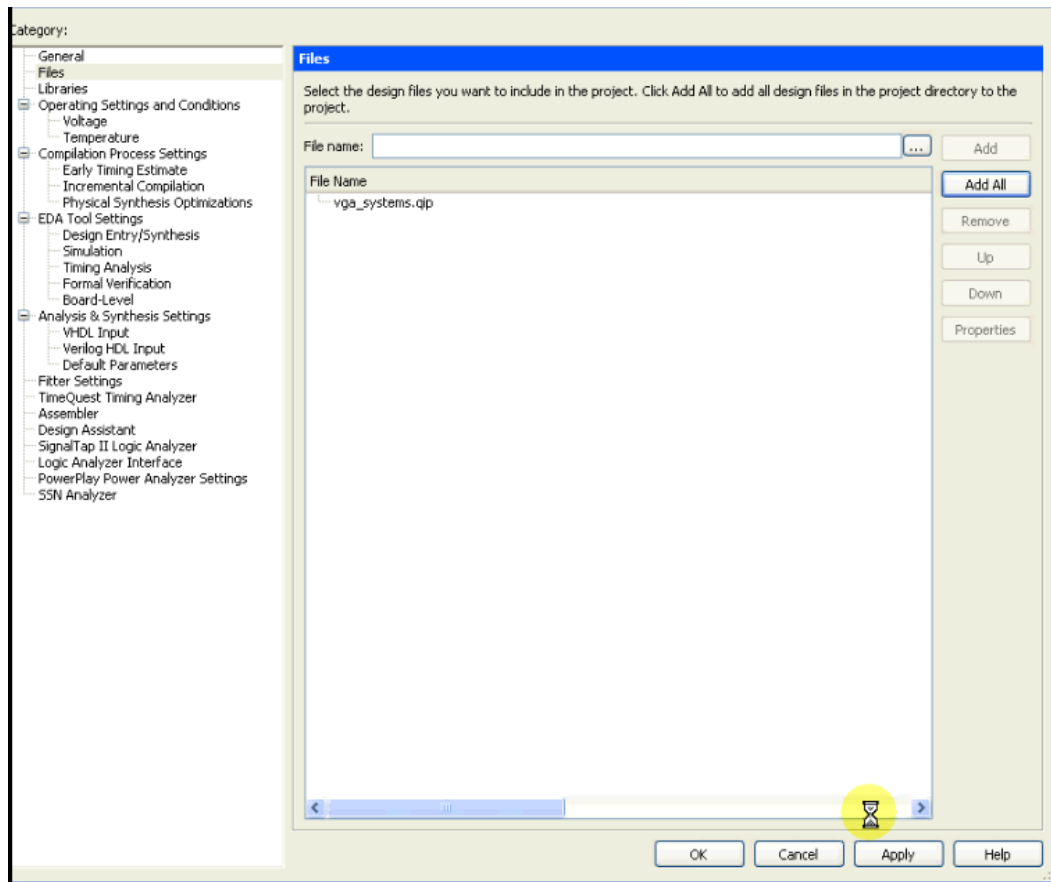
✓		cpu	Nios II Processor	[clk]				
		instruction_master	Avalon Memory Mapped Master	sys_clk				
		data_master	Avalon Memory Mapped Master	[clk]				
		jtag_debug_module	Avalon Memory Mapped Slave	[clk]	0x10400800	0x10400fff	IRQ 0	IRQ 31
✓		sram	SRAM/SSRAM Controller	[clock_reset]				
		avalon_sram_slave	Avalon Memory Mapped Slave	sys_clk	0x10200000	0x103fffff		
✓		sdram	SDRAM Controller	[clk]				
		s1	Avalon Memory Mapped Slave	sys_clk	0x08000000	0x0fffffff		
✓		jtag_uart	JTAG UART	[clk]				
		avalon_jtag_slave	Avalon Memory Mapped Slave	sys_clk	0x10401010	0x10401017		
✓		external_clocks	Clock Signals for DE-Series Board Peri...					
		avalon_clocks_slave	Avalon Memory Mapped Slave	clk_50	0x10401018	0x10401019		
✓		vga_pixel_buffer	Pixel Buffer DMA Controller	[clock_reset]				
		avalon_pixel_dma_ma...	Avalon Memory Mapped Master	sys_clk				
		avalon_control_slave	Avalon Memory Mapped Slave	[clock_reset]	0x10401000	0x1040100f		
		avalon_pixel_source	Avalon Streaming Source	[clock_reset]				
✓		video_rgb_resampler	RGB Resampler	[clock_reset]				
		avalon_rgb_sink	Avalon Streaming Sink	sys_clk				
		avalon_rgb_source	Avalon Streaming Source	[clock_reset]				
✓		video_scaler	Scaler	[clock_reset]				
		avalon_scaler_sink	Avalon Streaming Sink	sys_clk				
		avalon_scaler_source	Avalon Streaming Source	[clock_reset]				
✓		dual_clock_fifo	Dual-Clock FIFO					
		avalon_dc_buffer_sink	Avalon Streaming Sink	sys_clk				
		avalon_dc_buffer_so...	Avalon Streaming Source	vga_clk				
✓		video_vga_controller...	VGA Controller	[clock_reset]				
		avalon_vga_sink	Avalon Streaming Sink	vga_clk				



Εικόνα 17. SOPC Builder

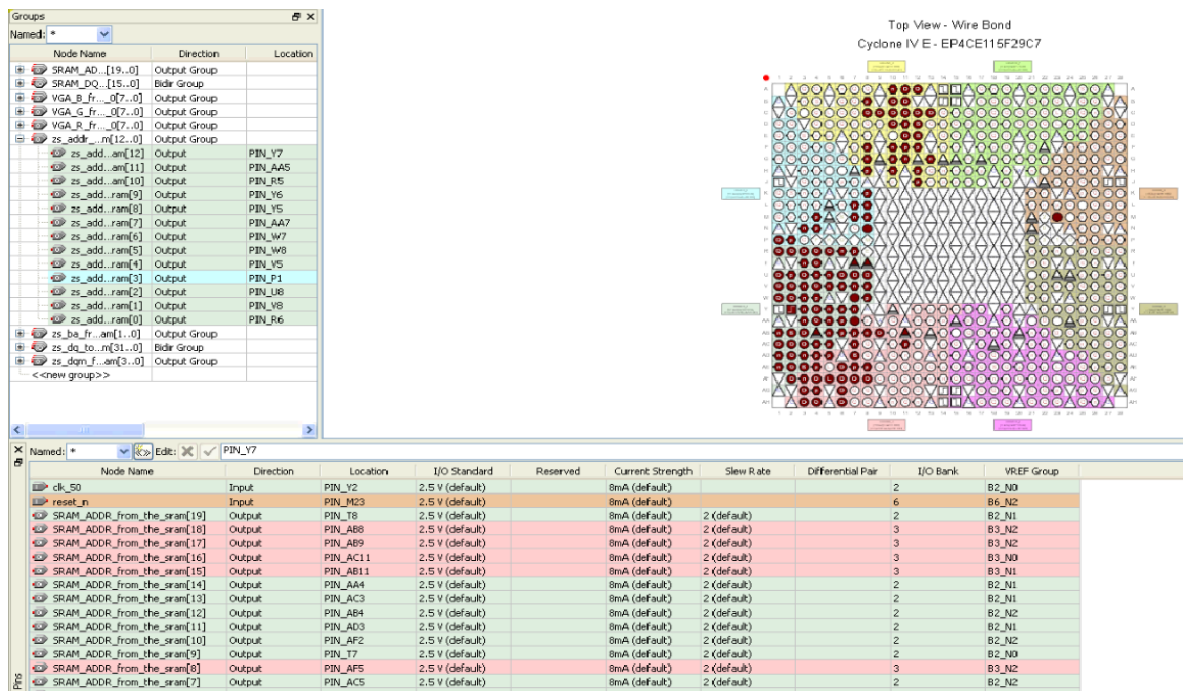
2.3 ΕΝΤΑΞΗ ΤΟΥ NIOS II ΣΥΣΤΗΜΑΤΟΣ ΣΤΟ QUARTUS II

Έχοντας ολοκληρώσει πλέον το σύστημα μένει να το εντάξουμε στο Quartus II. Πηγαίνουμε **Assignments > Settings** και μας εμφανίζεται το παράθυρο των ρυθμίσεων. Διαλέγουμε στις κατηγορίες (**Category**) τους φάκελους (**files**) και αφού επιλέξουμε όλα τα αρχεία τα αφαιρούμε πατώντας το **remove**. Στο **file name** πατάμε στο εικονίδιο της περιήγησης. Στο παράθυρο που εμφανίζεται επιλέγουμε στο **Files of type** το **Script files** και επιλέγουμε το αρχείο **vga_systems.qip**. Πατάμε **open**, μετά **add** και **ok**. Στη συνέχεια πηγαίνουμε **Processing > Start > Start Analysis & Elaboration**. Το πρόγραμμα ξεκινάει την προετοιμασία για την ενσωμάτωση του συστήματος, αυτό μπορεί να διαρκέσει λίγα λεπτά. Όταν ολοκληρωθεί θα παρουσιαστεί το μήνυμα **“Analysis & Elaboration was successful”**. Για να συνεχίσουμε πατάμε **ok**.



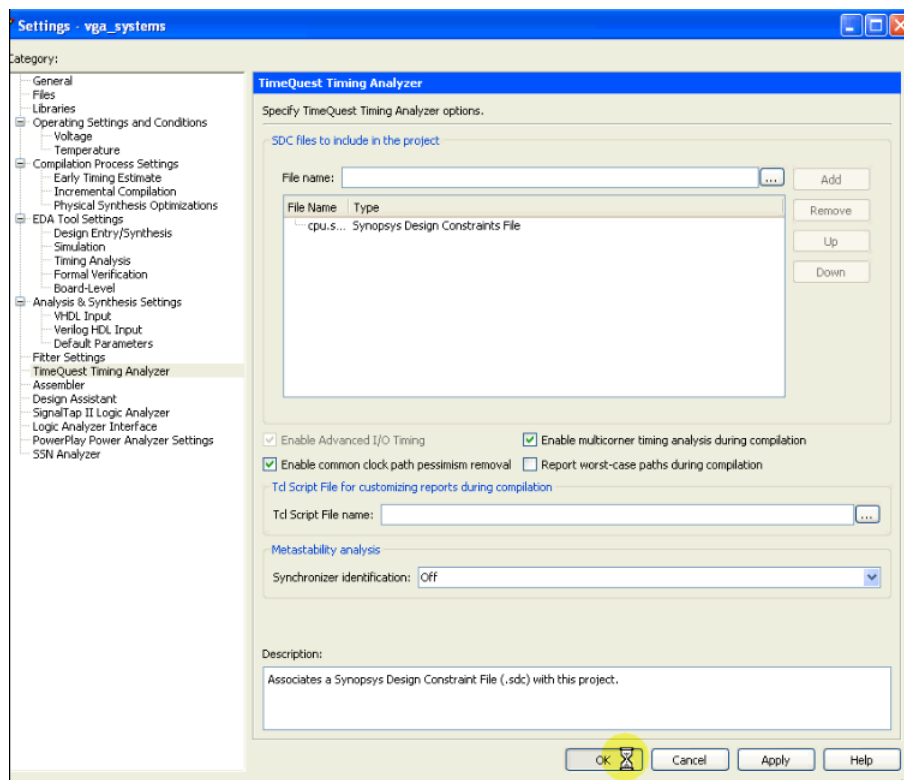
Εικόνα 18. Settings-files

Τώρα είμαστε έτοιμοι να συνδέσουμε τα PIN της πλακέτας. Για να το κάνουμε αυτό πηγαίνουμε **Assignments > Pin Planner**. Στο κάτω μέρος του παραθύρου που ανοίγει υπάρχει μια λίστα με όλα τα PIN από τις διεπαφές που είχαμε προσθέσει πριν στο SOPC Builder. Θα πρέπει για κάθε ένα από αυτά να βάλουμε στο **Location** την σωστή θέση πάνω στο τσιπ που φαίνεται πάνω δεξιά στην οθόνη. Αυτό θα το κάνουμε με τη βοήθεια του User Manual της DE2-115 που παρέχετε στο διαδίκτυο από την ALTERA. Αφού τελειώσουμε με όλα τα PIN πηγαίνουμε **File > Close** για να αποθηκευτούν οι αλλαγές. Στη συνέχεια πηγαίνουμε στο μενού **Assignments > Device**. Κάνουμε κλικ το **Device and Pin Options** και μετά διαλέγουμε το **Unused Pins**. Στο **Reserve all unused pins** επιλέγουμε το **As input tri-stated with weak pull-up** και πατάμε **ok** για να κλείσουν όλα τα παράθυρα.



Εικόνα 19. Pin Planner

Έπειτα πηγαίνουμε **Assignments > Settings** και στο **Category** επιλέγουμε **Time Quest Timing Analyzer**. Στο **file name** πατάμε στο εικονίδιο της περιήγησης. Βρίσκουμε τον φάκελο που δημιουργήσαμε για να αποθηκεύσουμε το project και επιλέγουμε το αρχείο **cpu.sdc**. Το ανοίγουμε πατάμε **add** και μετά **ok**.

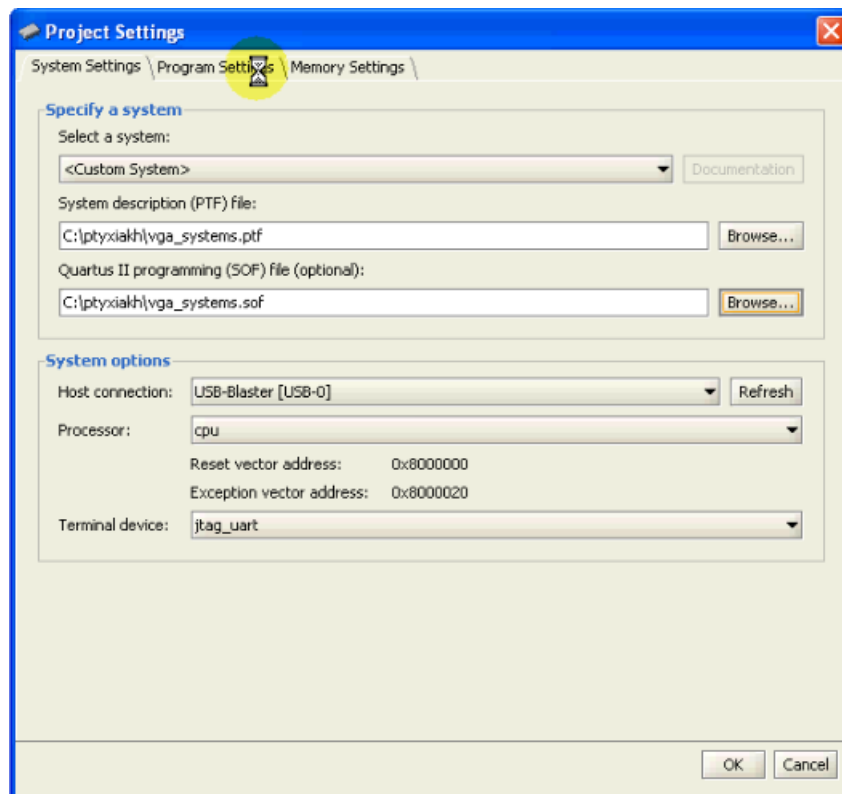


Εικόνα 20. Settings-Time Quest Timing Analyzer

Για να ολοκληρωθεί το project δε μένει παρά να το τρέξουμε. Για να γίνει αυτό πηγαίνουμε στο **Assignments > Start Compilation**. Αυτό θα χρειαστεί μερικά λεπτά για να ολοκληρωθεί. Όταν τελειώσει θα εμφανιστεί το μήνυμα “**Full Compilation was Successful**”. Με αυτό τον τρόπο έχουμε δημιουργήσει το vga_systems.ptf αρχείο που είναι απαραίτητο για να τρέξουμε το πρόγραμμά που θα εισάγουμε στο FPGA.

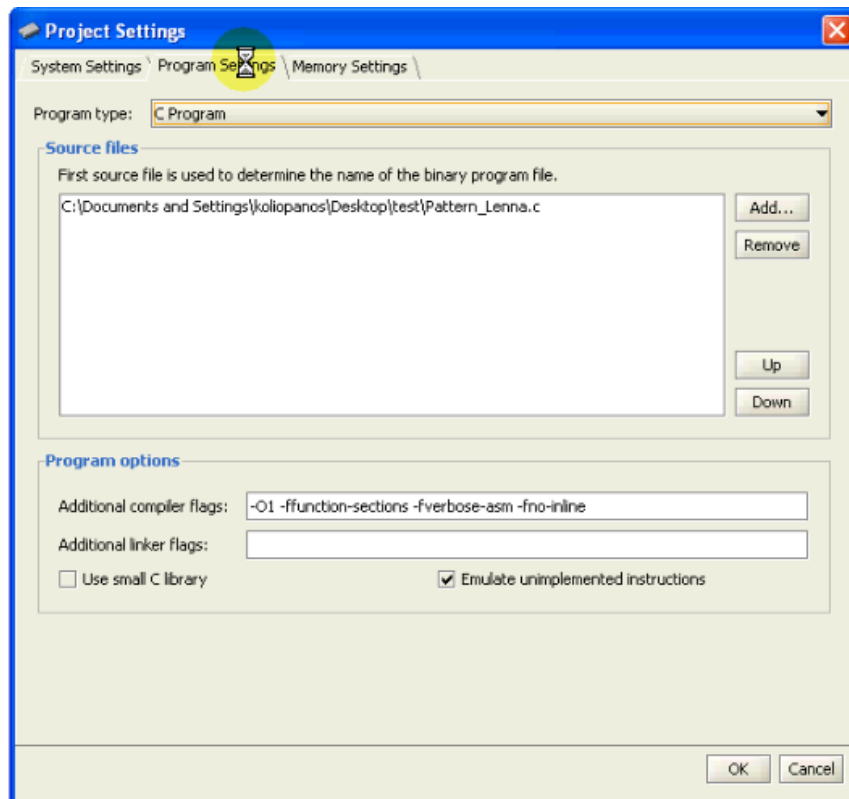
2.4 ΕΙΣΑΓΩΓΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΕ ΤΟ ALTERA MONITOR PROGRAM

Είμαστε έτοιμοι να τρέξουμε το πρόγραμμα στο FPGA. Ανοίγουμε λοιπόν το **ALTERA Monitor Program**. Πηγαίνουμε **File > New Project** και δημιουργούμε ένα νέο project. Στην ερώτηση που μας γίνεται πατάμε Yes και περιμένουμε να το φορτώσει στην πλακέτα. Θα εμφανιστεί το μήνυμα “**The SOPC Builder has been successfully downloaded on to the board**”. Πηγαίνουμε στο **Project Settings**, στην καρτέλα **System Settings**, στο **System description (PTF) File** θα επιλέξουμε το **vga_systems.ptf** από τον φάκελο που αποθηκεύονται τα αρχεία μας και στο **Quartus II Programming (SOF) File** επιλέγουμε το αρχείο **vga_systems.sof**.



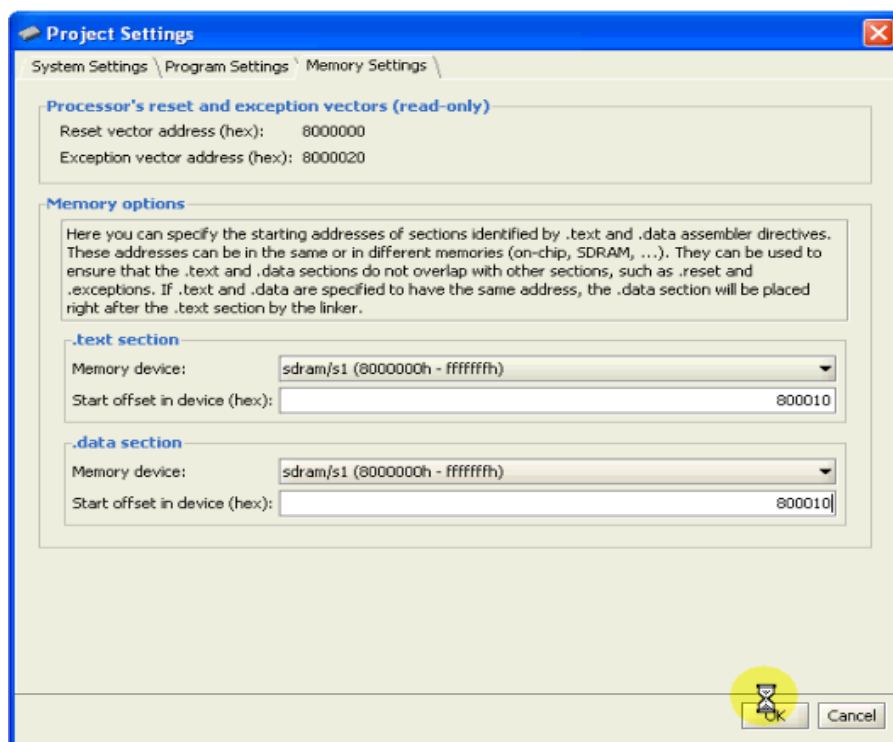
Εικόνα 21. Project Settings-1

Στην καρτέλα **Program Setting** πρέπει να επιλέξουμε το αρχείο με τον κώδικα στον οποίο θέλουμε να κατεβάσουμε και να τρέξουμε στο FPGA. Στη δικιά μας περίπτωση ο κώδικας μας δίνεται έτοιμος. Ο συγκεκριμένος κώδικας περιέχει ένα πλήθος από pixelστα οποία περιγράφουν την εικόνα την οποία θέλουμε να αναπαραστήσουμε.



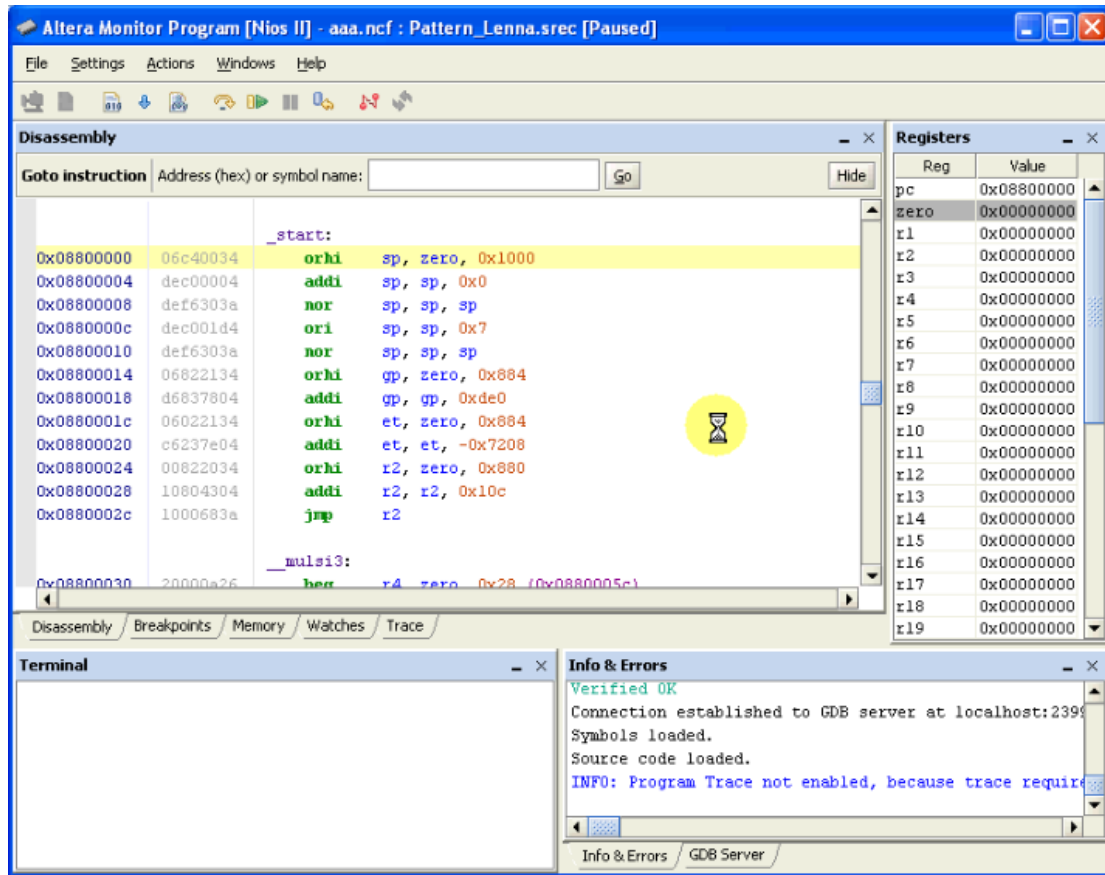
Εικόνα 22. Project Settings-2

Τέλος στην καρτέλα **Memory Settings**, στο *text Section*, στο **Start offset in device (hex)** πρέπει να βάλουμε την τιμή της διεύθυνσης της μνήμης η οποία φαίνεται από πάνω στο **Memory device**. Το ίδιο ακριβώς κάνουμε και στο **data section**, στο **Start offset in device (hex)** και πατάμε **ok**.



Εικόνα 23. Project Settings-3

Έχοντας κάνει όλες τις ρυθμίσεις μένει να τρέξουμε το πρόγραμμα. Πηγαίνουμε λοιπόν **Actions > Compile & Load**. Βλέπουμε το πρόγραμμα να τρέχει και να εμφανίζονται τα αποτελέσματα όπως στην εικόνα 24.



Εικόνα 24. Altera Monitor Program

Η εικόνα η οποία θα εμφανιστεί στην οθόνη που είναι συνδεδεμένη με το FPGA είναι το αποτέλεσμα της αναπαράστασης όλων των pixel τα οποία έχουν δοθεί με τον κώδικα, όπως αναφέρεται και παραπάνω. Το αποτέλεσμα φαίνεται και στην εικόνα 25.



Εικόνα 25. Η εικόνα lenna.c

ΠΗΓΕΣ - ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] : ftp://ftp.altera.com/up/pub/Altera_Material/12.1/Boards/DE2-115/DE2_115_User_Manual.pdf
- [2] : http://en.wikipedia.org/wiki/Quartus_II
- [3] : <http://www.altera.com/education/training/trn-index.jsp>
- [4] : Σχεδίαση Ψηφιακών Συστημάτων με τη Γλώσσα VHDL, Stephen Brown, Zvonko Vranesic.