



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Ενσωματωμένα Συστήματα

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο 12

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.icte.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα

1. Σκοπός της άσκησης	4
2. Παραδοτέα	4
3. Κατασκευή του System on Chip.....	5
4. Κατασκευή του Software.....	8

1. Σκοπός της άσκησης

- Προσθήκη ενός επιταχυντή IP υπολογισμού των τριγωνομετρικών συναρτήσεων ημίτονου και συνημιτόνου.
- Κατασκευή του υλικού (*System on Chip*) και του λογισμικού (*ρουτίνα εξυπηρέτησης διακοπής*).

(A) 3 ερωτήσεις

(C) 2 ασκήσεις/προγράμματα

4 bonus

2. Παραδοτέα

- **Παραδοτέο A1:** Screenshot
- **Παραδοτέο A2:** Screenshot από το ISE με το project
- **Παραδοτέο A3:** Screenshot με το σχηματικό και κείμενο μιας παραγράφου για το ποσοστό ολοκλήρωσης και τα προβλήματα που αντιμετωπίσατε.
- **Παραδοτέο C1:** αρχεία VHDL
- **Παραδοτέο C2:** Όλα τα αρχεία του project

Κατά το σχεδιασμό ενσωματωμένων συστημάτων, υπάρχουν περιπτώσεις που απαιτείται η εκτέλεση κάποιας λειτουργίας που είτε δεν υποστηρίζεται από τον ενσωματωμένο επεξεργαστή που έχουμε επιλέξει, είτε η εκτέλεση αυτής της λειτουργίας με υλοποίηση λογισμικού θα ήταν πολύ πολύπλοκη και χρονοβόρα. Σε αυτές τις περιπτώσεις μπορούμε να προσθέσουμε έναν επιταχυντή αυτής της λειτουργίας, δηλαδή μια μονάδα υλικού με αποκλειστική λειτουργία την εκτέλεση αυτής της συνάρτησης.

Σε αυτό το εργαστήριο θα προσθέσουμε σε ένα ενσωματωμένο σύστημα που βασίζεται στον επεξεργαστή riscoblaze, έναν επιταχυντή IP υπολογισμού ημίτονου και συνημιτόνου, που βασίζεται στον αλγόριθμο Cordic (*υλοποίηση με ολισθήσεις και προσθέσεις, χωρίς ανάγκη για πολλαπλασιαστές*).

Θα χρησιμοποιήσουμε το IP Core που βρίσκεται στη σελίδα (<http://www.ht-lab.com/freecores/cordic/cordic.html>) και έχει ανέβει στα έγγραφα του εργαστηρίου (**cordic.zip**).

(BONUS1 Υλοποίηση όλου του εργαστηρίου σε ALTERA DE2 +0.75 μονάδα)

3. Κατασκευή του System on Chip

- Κατεβάστε το αρχείο με το **cordic IP** και εξάγετε τα αρχεία σε έναν κατάλογο εργασίας. Από τα έγγραφα του εργαστηρίου, δείτε τα αρχεία `cordic_schematic_small.gif` και `cordic_schematic_large.gif` τα οποία είναι τα σχηματικά μοντέλα του συγκεκριμένου αλγορίθμου. Ανοίξτε και μελετήστε τα VHDL αρχεία που περιλαμβάνονται στο IP cordic.

→ *** Το συγκεκριμένο IP μπορεί να παραμετροποιηθεί να δέχεται είσοδο από 8 bit έως 32bit και να εξάγει το **sin()** και το **cos()** σε αντίστοιχο bitwidth. Εμείς θα χρησιμοποιήσουμε αρχικά 8bit, ώστε να μπορούν να μεταφερθούν σε μια περίοδο ρολογιού τα 8bit από την έξοδο του επεξεργαστή ricoblaze στο IP cordic. Αν θελήσουμε να μεταφέρουμε 32bit τότε θα έπρεπε να χρησιμοποιήσουμε έναν επιπρόσθετο καταχωρητή ολίσθησης 32bit, ο οποίος θα φόρτωνε 8bit σε κάθε περίοδο. Έτσι, σε 4 κύκλους ρολογιού θα είχε φορτώσει την είσοδο των 32bit. Αντίστοιχα, η έξοδος του IP cordic θα τοποθετούνταν σε έναν καταχωρητή των 32bit, ο οποίος σε κάθε περίοδο θα έστελνε τα 8bit στην 8bit είσοδο του επεξεργαστή. ***

- Πριν χρησιμοποιήσουμε το Cordic θα πρέπει να το προσομοιώσουμε για να επιβεβαιώσουμε την ορθή λειτουργία. Χρησιμοποιώντας το πρόγραμμα `gen_testvector.exe` που περιλαμβάνεται, δημιουργήστε τα διανύσματα εξόδου για 8 bit, τα οποία θα αποθηκευτούν στο αρχείο που θα ορίσετε εσείς.

→ *** Λίγα λόγια για τα διανύσματα που δημιουργούνται: Το cordic δέχεται ως είσοδο ένα όρισμα σε ακτίνια από $-\pi/2$ (-90° ή *ισοδύναμα* $3\pi/2$) έως $+\pi/2$ ($+90^\circ$). Με την εκτέλεση του προγράμματος δημιουργίας των διανυσμάτων ελέγχου για όλες τις μοίρες από -90 έως $+90$ γίνονται οι παρακάτω υπολογισμοί:

- Για μια δεδομένη τιμή σε ακτίνια [π.χ. 0.17 ή 10°] που δίνεται σε δεκαδική μορφή υπολογίζεται η αντίστοιχη τιμή στο δυαδικό σύστημα για το εύρος bit που έχουμε δηλώσει, σε μορφή σταθερής υποδιαστολής [00.001011 ($0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} = 0.171875$)]. Το MSB 0 είναι το πρόσημο, ενώ το επόμενο ψηφίο 0 είναι το significant (δηλαδή, αναπαρίστανται τιμές από -1.999 έως $+1.999$). Ο αριθμός 00.001011 αναπαριστάται στο δεκαεξαδικό σύστημα ως [0B]. Το $\sin()$ για 0.17 ακτίνια ή 10° έχει τιμή 0.1736 που έχει αναπαράσταση 0.001011, δηλαδή [0B], ενώ το $\cos()$ για 0.17 ακτίνια ή 10° έχει τιμή 0.984807753008 με αναπαράσταση 00.111111 = 0.984375 με δεκαεξαδική απεικόνιση [3F]. Για αυτό θα δούμε το πρόγραμμα δημιουργίας των διανυσμάτων να εκτυπώνει:

Angle=0.175[10] [0000000B] sin=0.174 [0000000B] cos=0.985 [0000003F]

ενώ στο αρχείο θα τοποθετηθεί η γραμμή για τη γωνία, το $\sin()$ και το $\cos()$ στο δυαδικό σύστημα:

10100111 11000001 00001011

Αν ο αριθμός είναι αρνητικός τότε χρησιμοποιείται το συμπλήρωμα ως προς 2. Για παράδειγμα, ο αριθμός -0.087 αναγράφεται ως 11111011 και στο δεκαεξαδικό σύστημα ως [FB]. Ο αριθμός αυτός εξάγεται ως εξής: Το 0.087 έχει αναπαράσταση 00.000101 . Ο αρνητικός αριθμός (-0.087) είναι το συμπλήρωμα ως προς 2 του παραπάνω αριθμού, δηλαδή η αντιστροφή του παραπάνω αριθμού (11.111010) και η πρόσθεση του 1 (11.111011). Το πρόγραμμα δημιουργίας των διανυσμάτων θα εκτυπώσει:

Angle=-0.087[-5] [000000FB].....

ενώ στο αρχείο θα τοποθετηθεί η γραμμή:

11111011

- Με τον ίδιο τρόπο εξάγονται τα διανύσματα για οποιοδήποτε μήκος bit. Τα διανύσματα αυτά χρησιμοποιούνται για την ορθή επαλήθευση του κυκλώματος. Δηλαδή, διαβάζεται μια γραμμή από τον προσομοιωτή, εισάγεται ως είσοδος στο cordic το διάνυσμα της πρώτης στήλης, και συγκρίνονται τα αποτελέσματα από τις εξόδους **sin()** και **cos()** κατά την προσομοίωση του IP **cordic** με τη δεύτερη και τρίτη στήλη αντίστοιχα. Αν είναι ίδια τα αποτελέσματα, τότε η προσομοίωση είναι επιτυχής για το συγκεκριμένο διάνυσμα εισόδου.
- Ανοίξτε το modelsim.
- τροποποιήστε το path για το vector file που έχετε δημιουργήσει προηγουμένως, στα αρχεία **cordic_tb_struct.vhd** και **cordic_tester_behavioral.vhd** (σε 3 γραμμές).
- κάντε compile όλα τα VHDL αρχεία με επισήμανση να τοποθετηθούν στη **βιβλιοθήκη cordic** (διαφορετικά τροποποιήστε τα αρχεία στις πρώτες γραμμές, αντί για *Library cordic*, να δηλώνουν το *Library work*) για **bitwidth=8bit** (τροποποιήστε κατάλληλα το *generic decleration*).
- προσομοιώστε το **cordic_tb (struct)**. Εκτελέστε για μερικούς κύκλους μέχρι να εμφανιστεί το πρώτο αποτέλεσμα:
(δεν είναι σωστό το U-unassigned ή 0).
- Μεγιστοποιήστε το παράθυρο των κυματομορφών στις οποίες να φαίνονται οι εξοδοί **sin()** και **cos()** και οι τιμές που έχουν και αποτυπώστε το screenshot.

Παραδοτέο A1: Screenshot

Αφού γίνει η προσομοίωση και επιβεβαιωθεί η ορθή λειτουργία σύμφωνα με το αρχείο των διανυσμάτων που έχουμε δώσει, μπορούμε να εισάγουμε το IP στο κύκλωμά μας. Συγκεκριμένα:

- Δημιουργήστε ένα νέο project στο Xilinx ISE.
- Εισάγετε το αρχείο vhdl του επεξεργαστή picoblaze.
- Εισάγετε τα αρχεία της VHDL για το IP CORDIC στο project (όχι τα αρχεία για το *testbench*, δε χρειάζεται).
- Δημιουργήστε ένα top level design (χρησιμοποιήστε το από προηγούμενο εργαστήριο), στο οποίο έχετε προσθέσει και δηλώσει το component cordic με generic width=8
- Στο top level design, θέλετε να χρησιμοποιήσετε επίσης τα buttons, τα leds και το LCD screen (να υπάρχουν δηλωμένα, όπως και οι κατάλληλες διασυνδέσεις από το προηγούμενο εργαστήριο).
- Να προστεθούν τα κατάλληλα signals για τη λειτουργία του cordic component.
- Να τοποθετηθεί **μια process με cordicIP: cordic port map (...)**

Στο port map θα πρέπει να συνδέσετε κατάλληλα την 8bit έξοδο του επεξεργαστή στην 8bit είσοδο του cordic, όταν επιλέγεται η κατάλληλη port εξόδου (επιλέξτε εσείς μια, με παρόμοιο τρόπο όπως υπάρχει για το LCD ή για τα LED).

- Θα πρέπει να τροποποιήσετε το `interrupt_control: process` με τη διαγραφή των αχρείαστων καταστάσεων, και την εισαγωγή μιας κατάστασης στην οποία όταν ολοκληρωθεί ο υπολογισμός από το cordic (δηλαδή `done=1`) τότε `interrupt<-1`.
- Τροποποίηση `input_ports`: Αύξηση του εύρους του επιλογέα κατά 1 bit (δηλαδή αντί για 2bit να είναι 3 bit), ώστε να προστεθούν και άλλες καταστάσεις:
- αν **0x03** τότε στο `in_port` να συνδεθεί η έξοδος `sin()` του IP.
- αν **0x04** τότε στο `in_port` να συνδεθεί η έξοδος `cos()` του IP.
- Τροποποίηση `output_ports` ώστε:
- αν το **bit** (4) είναι ενεργοποιημένο να προκαλεί ένα reset στο cordic (αποστολή '1' σε κατάλληλο port).

- αν το **bit** (5) είναι ενεργοποιημένο να τοποθετεί τα δεδομένα του out_port[8:0] του επεξεργαστή στην είσοδο angle[8:0] του cordic και να θέτει το '1' στο start του cordic.
- Εισάγετε ένα οποιοδήποτε αρχείο ROM για να γίνει compile, το οποίο θα το αντικαταστήσετε στην επόμενη παράγραφο.
- Κάντε compile και επιβεβαιώστε ότι όλα έχουν συνδεθεί. Δείτε το schematic.

Παραδοτέο A2: Screenshot από το ISE με το Project ανοιχτό στο οποίο φαίνεται το top level design, και από κάτω η μνήμη ROM, ο επεξεργαστής, και το Cordic.

Παραδοτέο A3: Screenshot με το σχηματικό

Παραδοτέο C1: Τα αρχεία VHDL που χρησιμοποιήσατε

4. Κατασκευή του Software

- Κατασκευάστε κατάλληλο rsm αρχείο assembly για τον επεξεργαστή, ο οποίος θα στέλνει για υπολογισμό μια συγκεκριμένη hardcoded τιμή (επιλέξτε μια από το vector file) στο Cordic , στη συνέχεια θα καλεί μια συνάρτηση καθυστέρησης και θα πηγαίνει σε μια ετικέτα πριν την κλήση της καθυστέρησης σε έναν ατέρμονο βρόχο.
- Θα πρέπει να κατασκευάσετε μια συνάρτηση, η οποία θα είναι η συνάρτηση χειρισμού διακοπής για το Cordic. Δείτε τον κώδικα από το προηγούμενο εργαστήριο και θα δείτε στο τέλος του rsm αρχείου, υπάρχει η δήλωση μιας διεύθυνσης μνήμης (**ADDRESS 3FF**) που σημαίνει η επόμενη εντολή να γραφεί στη διεύθυνση 3FF και μετά η εντολή (**JUMP ISR**). Η διεύθυνση 3FF είναι η διεύθυνση που θα εκτελέσει ο picoblaze όταν γίνει κάποιο interrupt. Εσείς θα γράψετε λοιπόν κώδικα assembly για τη συνάρτηση χειρισμού διακοπής ISR (interrupt service request) που θα κάνει τα παρακάτω:
- Θα διαβάζει την τιμή από την έξοδο sin() του IP cordic
- Θα εκτυπώνει την 8bit δυαδική τιμή στο LCD και θα περιμένει το πάτημα ενός οποιουδήποτε πλήκτρου.
- Μόλις πατηθεί κάποιο πλήκτρο θα διαβάζει την τιμή από την έξοδο cos() του IP cordic.
- Θα εκτυπώνει την 8bit δυαδική τιμή στο LCD και θα περιμένει το πάτημα ενός οποιουδήποτε πλήκτρου.
- θα στέλνει το σήμα Reset στο Cordic.

- Από το PSM αρχείο, εξάγετε το rom instruction memory vhdl αρχείο, όπως έχει γίνει σε προηγούμενο εργαστήριο, αντικαταστήστε το παλαιό rom vhdl αρχείο και δημιουργήστε το νέο bitstream.
- Προγραμματίστε την πλακέτα με το νέο bitstream και επιβεβαιώστε την ορθή λειτουργία

Παραδώστε όλα τα αρχεία του project **(C2)**, μαζί με ένα κείμενο μιας παραγράφου για το ποσοστό ολοκλήρωσης και τα προβλήματα που αντιμετωπίσατε **(A3)**.

Μελλοντικές επεκτάσεις του εργαστηρίου:

- υποστήριξη 32bit **(BONUS2 +0.75 μονάδα):**
- Χρησιμοποιείται είσοδος 32bit .
- Διαβάζεται η έξοδος 32bit και εισάγεται στο 8bit επεξεργαστή .
- Χρησιμοποιείται το rotary control για να κάνει scroll τα 32 bits.
- Χρήση interrupt για χρήση 2 cordic **(BONUS3 +0.75 μονάδα).**
- Χρησιμοποιείται ένα δεύτερο cordic, ώστε να μπορεί να γίνει παράλληλα ο υπολογισμός 2 γωνιών.
- Το Interrupt ενεργοποιείται από 2 καταστάσεις. Όταν συμβεί interrupt ο επεξεργαστής πρέπει να κάνει rolling για να διαπιστώσει πια συσκευή έστειλε το interrupt.
- Χρησιμοποιείται η UART θύρα, ώστε να δέχεται την είσοδο της γωνίας μέσω σειριακής θύρας **(BONUS4 +0.75 μονάδα).**

***** Μέγιστο BONUS εργαστηρίου: 1.5 μονάδα *****