5ο Εργαστήριο Αρχιτεκτονικής Η/Υ: Σχεδίαση κυκλωμάτων με το Quartus

Α. Ευθυμίου

Παραδοτέο: Παρασκευή 3 Απρίλη, 23:00

Το αντικείμενο αυτής της άσκησης είναι η εξοικίωση με το εργαλείο Quartus για σχεδίαση κυκλωμάτων και το ModelSim-Altera (Starter Edition) για προσομοίωση. Και τα δύο θα χρησιμοποιηθούν σε επόμενες εργαστηριακές ασκήσεις για σχεδίαση επεξεργαστών, οπότε ο χρόνος που θα καταναλώσετε σε αυτή την άσκηση θα διευκολύνει σημαντικά στις επόμενες ασκήσεις.

Σε αυτή την εργαστηριακή άσκηση δεν χρειάζεται να μελετήσει κανείς το σύγγραμμα. Πρέπει όμως να θυμάστε τη γλώσσα Verilog και βασικές αρχές ψηφιακής σχεδίασης.

Μή ξεχάσετε να επιστρέψετε τα παραδοτέα που αναφέρονται στο τέλος του κειμένου για να πάρετε βαθμό γι'αυτή την εργαστηριακή άσκηση!

Για οδηγίες εγκατάστασης του Quartus, δείτε το φυλλάδιο εργαστηρίου 0.

Η μορφή αυτής της άσκησης είναι μια μεγάλη ακολουθία από βήματα που θα κάνετε για να σχεδιάσετε και να προσομοιώσετε ένα μικρό κύκλωμα. Χρειάζεται προσοχή στις λεπτομέρειες (επιλογές μενού κλπ) γιατί μικρά λάθη απροσεξίας δημιουργούν αργότερα προβλήματα που είναι δύσκολο να διαγνωστούν και να αντιμετωπιστούν. Σε διάφορα σημεία υπάρχουν εξηγήσεις διαφόρων όρων που χρησιμοποιούν τα εργαλεία και άλλα σχόλια. Η διαδικασία σχεδίασης και προσομοίωσης είναι παρόμοια για άλλα κυκλώματα και θα την ακολουθήσετε σε επόμενες ασκήσεις.

1 Εκκίνηση Quartus

Ξεκινήστε το Quartus σύμφωνα με το σύστημά σας. Στους υπολογιστές του εργαστηρίου σε ένα τερματικό δώστε τις παρακάτω εντολές. Η πρώτη εντολή θέτει διάφορες μεταβλητές περιβάλοντος που είναι απαραίτητες για να τρέξει το Quartus. Αν χρησιμοποιείτε προσωπικό υπολογιστή, θα πρέπει να αλλάξετε τα μονοπάτια αρχείων κατάλληλα.

```
source ~myy402/env/quartus.env
quartus &
```

Θα δείτε ένα μεγάλο παράθυρο και μία splash-screen που παραμένει στην οθόνη. Κλείστε την splash-screen πατώντας το X πάνω δεξιά.

2 Δημιουργία project

Για κάθε καινούρια εργασία στο Quartus χρειάζεται ένα project που περιλαμβάνει όλα τα αρχεία και τις ρυθμίσεις που απαιτεί η εργασία. Από το μενού διαλέξτε File > New Project Wizard Δώστε τον αρχικό κατάλογο του project (όποιον θέλετε) και το όνομα **tutorial** για το project που θα δημιουργήσετε. Πατήστε Next και στην επόμενη σελίδα ξανά Next γιατί δεν έχουμε, για την ώρα, να προσθέσουμε αρχεία στο project. Στην επόμενη σελίδα διαλέξτε Cyclone II και EP2C35F672C6, που βρίσκεται προς το τέλος του πίνακα και πατήστε Next[1]. Στη σελίδα που θα εμφανιστεί, αλλάξτε την τρίτη στήλη του Simulation σε Verilog. Η δεύτερη στήλη θα πρέπει να γράφει ModelSim-Altera. Αλλάξτε το αν δεν γράφει αυτό. Πατήστε ξανά Next και θα δείτε μια σύνοψη με τις επιλογές που κάνατε. Βεβαιωθείτε ότι είναι εντάξει και πατήστε Finish.

3 Σχεδίαση

Από το κεντρικό μενού, διαλέξτε File > New και μετά "Design Files" > BlockDiagram/SchematicFile από το παράθυρο που θα εμφανιστεί. Θα δείτε ότι το κεντρικό τμήμα του παραθύρου αλλάζει και

[1] Αυτό είναι το ολοκληρωμένο κύκλωμα των πλακετών DE2 που έχουμε στο εργαστήριο.

εμφανίζεται ένα tab με όνομα Block1.bdf. Αυτό είναι ένα σχηματικό όπου αργότερα θα σχεδιάσουμε πύλες και θα τοποθετήσουμε blocks που ομαδοποιούν είτε άλλες πύλες ή και κώδικα Verilog. Θα το χρησιμοποιήσουμε ώς το τελικό μας κύκλωμα, top-level design, στα Αγγλικά.

Για την ώρα αποθηκεύστε το με File > Save As. Αν στο παράθυρο που θα εμφανιστεί, το όνομα αρχείου δεν είναι το tutorial, αλλάξτε το σε αυτό. Επίσης βεβαιωθείτε ότι το check-box "Add file to current project" είναι επιλεγμένο. Πατήστε το Save.

Πατήστε ξανά File > New, αλλά αυτή τη φορά διαλέξτε "Verilog HDL File". Θα εμφανιστεί ένα καινούριο tab που μοιάζει με editor και με τίτλο Verilog1.v Αντιγράψτε τον παρακάτω κώδικα Verilog[2] στο παράθυρο αυτό. Παρατηρήστε ότι ο editor κάνει syntax-highlighting ώστε να ξεχωρίσουν οι δεσμευμένες λέξεις της Verilog, μεταβλητές, σταθερές κτλ.

```
module incrementer (
 input      [3:0] in,
 output reg [4:0] out
);

 always @(in)
     out = in + 4'h01;

endmodule
```

Το module που μόλις γράψατε είναι ένας incrementer, παρόμοιο κύκλωμα με τον αθροιστή αλλά με τον ένα προσθετέο πάντα ίσο με τον αριθμό 1. Παρατηρήστε ότι η έξοδος χρησιμοποιεί 1 επιπλέον bit γιατί μπορεί να γίνει υπερχείλιση.

Σώστε το αρχείο (File>Save ή Ctrl-S) αλλάζοντάς το όνομά του σε **incrementer.v**, αφού βεβαιωθείτε ότι το check-box "Add file to current project" είναι επιλεγμένο.

Για να τοποθετήσουμε το verilog module σε ένα σχηματικό διάγραμα, χρειάζεται να έχει ένα σύμβολο. Ευτυχώς το Quartus μπορεί να το κάνει αυτόματα για εμάς επιλέγοντας: File > Create/Update > Create Symbol Files for Current File. Αφού πάρετε το μήνυμα ότι το σύμβολο δημιουργήθηκε, κλείστε το νέο, άδειο Tab που εμφανίζεται.

Διαλέξτε τώρα το Tab tutorial.bdf, το σχηματικό που δημιουργήσατε στην αρχή.

Πατήστε, στο εσωτερικό παράθυρο το εικονίδιο που μοιάζει με πύλη AND (**symbol tool**). Με αυτό θα διαλέξουμε το σύμβολο που μόλις δημιουργήσαμε για να το τοποθετήσουμε στο σχηματικό. Θα εμφανιστεί ένα παράθυρο σαν αυτό του σχήματος 1. Πάνω αριστερά, κάτω από το Libraries: πατήστε το + δίπλα από το Project και θα δείτε το incrementer που μόλις δημιουργήσαμε και μετά πατήστε OK.

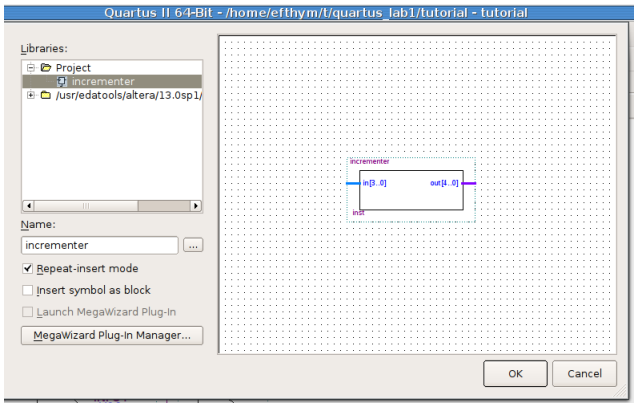
Το σύμβολο του incrementer ακολουθεί το δείκτη του ποντικιού. Διαλέξτε ένα κατάλληλο σημείο και κάντε κλικ για να τοποθετήσετε τον incrementer.

Επειδή συχνά τοποθετούμε πολλές πύλες σύμβολα, θα ξαναδείτε το σύμβολο να ακολουθεί το δείκτη. Πατήστε το Esc για να σταματήσει αυτό.

Ξαναπατήστε το symbol tool, αλλά αυτή τη φορά διαλέξτε την 2η βιβλιοθήκη που το ονομά της είναι στην ουσία το μονοπάτι όπου είναι εγκατεστημένο το Quartus. Μετά διαλέξτε primitives > logic > and2 για να πάρετε την πύλη AND 2 εισόδων. Τοποθετήστε 4 πύλες στα αριστερά του incrementor τη μία κάτω από την άλλη.

Είναι σημαντικό τα σχηματικά να είναι καλοσχεδιασμένα. Προσπαθήστε να τοπθετήσετε τις πύλες ώστε να είναι στοιχισμένες και σε ίση απόσταση μεταξύ τους. Μπορείτε να μετακινήσετε αντικείμενα στο σχηματικό, πιάνοντάς τα με το ποντίκι και σέρνοντας τα εκεί που θέλετε (αρκεί να έχετε επιλέξει το selection tool με εικονίδιο που μοιάζει με δείκτη ποντικιού). Υπάρχουν πολλές δυνατότητες σχεδίασης,

[2]Μπορείτε να βρείτε τον κώδικα για το εργαστήριο στο αποθετήριο lab05_starter GitHub.



Σχήμα 1: Το παράθυρο του symbol tool.

μετακίνησης αντικειμένων και τα συνήθισμένα: αποκοπή (cut), αντιγραφή (copy), επικόλληση (paste), αναίρεση (undo).

Για να προσθέσουμε εισόδους και εξόδους σε ένα σχηματικό, χρησιμοποιούμε το **pin tool**, ένα εικονίδιο που βρίσκεται στα δεξιά του symbol tool. Κάντε κλικ στο βέλος του pin tool και διαλέξτε input, ώστε να τοποθετήσουμε ακροδέκτες εισόδου για το κύκλωμα. Τοποθετήστε 4 pins ευθυγραμισμένα με την επάνω είσοδο κάθε AND αλλά σε κάποια απόσταση από αυτές, όπως στο σχήμα 2. Για να δόσετε ονόματα στα pins είτε κάντε διπλό-κλικ στο υπάρχον όνομα (π.χ. pin_name1) ή δεξί κλικ στο σύμβολο του pin και επιλέξτε properties (ιδιότητες). Αλλάξτε τα ονόματα σε a, b, c, d από επάνω προς τα κάτω.

Συνδέστε τα pins με τις απέναντι εισόδους των AND με καλώδια: Κάντε κλικ στο **Orthogonal Node Tool** που μοιάζει με ανάποδο κεφαλαίο Γ. Θα εμφανιστεί ένας σταυρός ως δείκτης ποντικιού και το σύμβολο του εργαλείου δίπλα του.

Κάντε κλικ στη «μυτερή», δεξιά μεριά του pin και, κρατώντας το κουμπί, τραβήξτε ώς απέναντι την είσοδο της AND. Δεν δουλεύει με κλικ στην αρχή και κλίκ στο τέλος της γραμμής, πρέπει να κρατάτε πατημένο το κουμπί του ποντικιού και να «τραβάτε» το καλώδιο.

Αφού συνδέσετε και τα τέσερα pins, σχεδιάστε ένα καλώδιο από την ελεύθερη είσοδο της επάνω AND, λίγο αριστερά και μέχρι λίγο κάτω από την τελευταία AND.

Προσθέστε ακόμα ένα pin εισόδου με όνομα r.

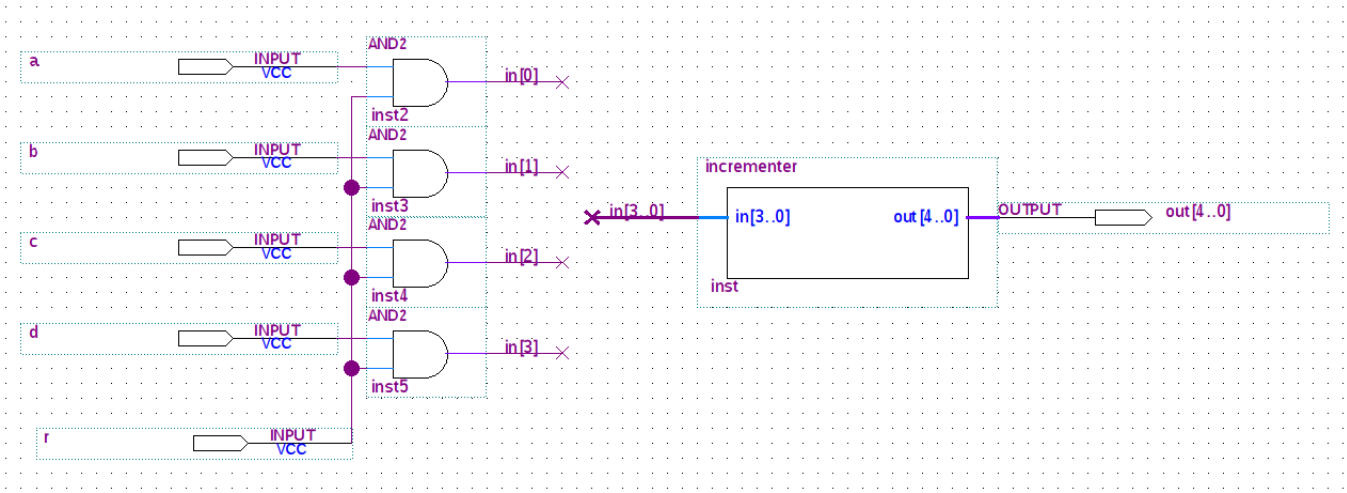
Θα πρέπει το κύκλωμά σας να αρχίσει να μοιάζει με το σχήμα 2. Για να σταματήσετε τη σχεδίαση καλωδίου πατήστε Esc ή διαλέξτε το εργαλείο selection tool με εικονίδιο ένα δείκτη οθόνης.

Παρόλο που το κατακόρυφο καλώδιο περνάει πάνω από τα άλλα, δεν συνδέονται μεταξύ τους. Όταν συνδέονται δύο καλώδια αυτό εμφανίζεται με ένα έντονο κύκλο στο σημείο της σύνδεσης.

Τώρα συνδέστε τις υπόλοιπες εισόδους των AND με το καλώδιο που συνδέεται στο pin r.

Αν κατά τη σχεδίαση κάνετε ένα λάθος τότε μπορείτε να διαλέξετε το σύμβολο ή το τμήμα καλωδίου που δεν τοποθετήθηκε σωστά και να το σβήσετε πατώντας το πλήκτο Delete (**όχι το Backspace**) ή από το μενού Edit > Delete. Μπορείτε επισης να αλλάξετε θέση σε σύμβολα επιλέγοντας και τραβώντας τα με το ποντίκι. Προσοχή όμως στα ήδη συνδεδεμένα σύμβολα γιατί η μετακίνηση μπορεί να προκαλέσει συνδέσεις καλωδίων που δεν θέλετε. Υπάρχει επίσης η δυνατότητα να αντιστρέψετε την τελευταία ενέργειά σας (Edit > Undo, Ctrl-Z).

Για να συνδέσουμε τις εξόδους των AND με την είσοδο του incrementer θα χρησιμοποιήσουμε ένα διαφορετικό τρόπο. Διαλέξτε το **Orthogonal Bus Tool** που μοιάζει με αυτό του καλωδίου που χρησιμοποιούσαμε αλλά έχει πιο χοντρή γραμμή (αλλά χωρίς λευκό κενό μέσα της). Ζωγραφίστε ένα **bus** από την είσοδο του incrementer μέχρι κάπου στη μέση της απόστασης από τις AND. Το bus είναι μια



Σχήμα 2: Το σχηματικό διάγραμμα tutorial.bdf

ομάδα καλωδίων που είναι παράλληλα.

Μετά κάντε δεξί κλικ πάνω στο bus, διαλέξτε properties και δώστε το όνομα `in[3..0]`. Αυτό το όνομα δηλώνει ότι το bus έχει 4 καλώδια από το 3 ως το 0.

Προσοχή μή χρησιμοποιείτε ονόματα που τελειώνουν σε αριθμό (π.χ. το `in1[3..0]`) για busses γιατί δημιουργούν προβλήματα. Επίσης η σύμβαση που χρησιμοποιείται στο σχηματικό για να δείξει πολλαπλά bit (και αντίστοιχα καλώδια) είναι διαφορετική από τη Verilog. Στη Verilog θα γράφαμε `in[3:0]` ενώ στο σχηματικό γράφουμε `in[3..0]`.

Τώρα σχεδιάστε καλώδια (Orthogonal Node Tool) από τις εξόδους των AND μέχρι κάπου κοντά στην αριστερή άκρη του προηγούμενου bus, αλλά χωρίς να ακουμπούν. Με επιλογή κάθε καλωδίου, δεξί κλικ και properties, δώστε στα καλώδια ονόματα: `in[0]` έως `in[3]` από πάνω προς τα κάτω. Το αποτέλεσμα θα πρέπει να μοιάζει με το σχήμα 2.

Τα ονόματα που δώσατε στα καλώδια αποτελούν τα επιμέρους bits του bus επειδή έχουν το ίδιο κυρίως όνομα (in). Έτσι, έχουμε πλέον συνδέσει τις εξόδους των AND με το bus και, συνεπώς, με την είσοδο του incrementor. Αν θέλαμε θα μπορούσαμε να σχεδιάσουμε τα καλώδια να ακουμπάνε στο bus για να φαίνεται καθαρά η σύνδεση. Αυτό όμως δεν είναι απαραίτητο. Σε ένα σχηματικό, καλώδια με το ίδιο όνομα θεωρούνται ότι συνδέονται. Αυτός είναι ένας τρόπος για να απλοποιούμε ένα σχέδιο που αλλιώς θα γινόταν πολύ πολύπλοκο.

Αν θέλαμε να συνδέσουμε τα επιμέρους καλώδια με το bus, θα φαινόταν καλύτερα αν σχεδιάζαμε μια προέκταση του bus που να πηγαίνει κατακόρυφα. Είναι σημαντικό πάντως ότι ακόμα και αν το σχεδιάζαμε έτσι, τα ονόματα στα καλώδια είναι απαραίτητα γιατί αλλιώς δεν θα μπορεί να γνωρίζει το Quartus σε πιο bit του bus συνδέεται το κάθε καλώδιο.

Αν στα σχηματικά σας έχετε καλώδια που δεν έχουν όνομα, στην προσομοίωση θα δείτε σήματα με παράξενα ονόματα που δημιουργούνται αυτόματα από το Quartus. Επειδή θα είναι δύσκολη η αντιστοίχιση, σας προτείνω να δίνετε ονόματα σε όλα τα καλώδια που πιστεύετε ότι θα σας ενδιαφέρει να παρατηρήσετε κατά την προσομοίωση.

Για να ολοκληρώσετε το σχηματικό, βάλτε ένα output pin και συνδέστε το απευθείας στην έξοδο του incrementor. Δώστε του το όνομα `out[4..0]`.

Οταν τελειώσετε με τη σχεδίαση, ελέγξτε και σώστε το σχηματικό.

Τώρα θα πρέπει να ζητήσετε από το Quartus να ελέγξει αν το κύκλωμά σας είναι εντάξει. Αυτό γίνεται με Processing > Start Compilation (ή με το εικονίδιο που μοιάζει με ένα τρίγωνο - playback).

Μετά από λίγη ώρα το Quartus θα σας ενημερώσει για το αποτέλεσμα. Θα δείτε πολλές πληροφορίες στο κάτω παράθυρο. Επίσης εμφανίζεται ένα καινούριο tab με πληροφορίες για το κύκλωμα. Ρίξτε μια ματιά και κλείστε το Compilation Report tab. Αν βρεθούν λάθη, θα εμφανιστούν γραμμές με κόκκινα γράμματα που εξηγούν το λάθος. Συνήθως με διπλό κλικ σε γραμμή λάθους, μεταβαίνετε στο αντικείμενο που έχει το πρόβλημα.

Ενα συνηθισμένο πρόβλημα είναι αντικείμενα (πύλες, pins, κ.α.) που είναι κατά λάθος τοποθετημένα ακριβώς το ένα πάνω από το άλλο, π.χ. 2 pins επειδή πατήσατε 2 φορές κλικ όταν τα τοποθετούσατε. Το αντικείμενο που βρίσκεται απο κάτω δεν φαίνεται αλλά υπάρχει και θα παίρνετε μηνύματα λάθους.

Φυσικά πρέπει να διορθώσετε τα λάθη πριν να συνεχίσετε. Αλλά μπορείτε να αγνοήσετε τις προειδοποιήσεις (warnings) γιατί σχετίζονται με βήματα σχεδίασης που δεν θα μας απασχολήσουν στο μάθημα.

Σε αυτό το σημείο το κύκλωμά σας είναι σωστό με την έννοια ότι δεν έχει βραχυκυκλώματα, ασύνδετες πύλες κλπ. Είναι το ισοδύναμο ενός προγράμματος που περνάει από τον μεταγλωτιστή. Δεν γνωρίζουμε ακόμη αν δουλεύει σωστά. Γι'αυτό θα πρέπει να τρέξουμε προσομοιώσεις: να δώσουμε εισόδους στο κύκλωμα και να ελέγξουμε αν οι έξοδοι παίρνουν τις σωστές τιμές.

Αυτή η διαδικασία ονομάζεται επαλήθευση (verification) και όχι έλεγχος ορθής λειτουργίας (testing) όπως συνηθίζεται σε λογισμικό. Ο λόγος είναι ότι ο όρος έλεγχος (testing) για hardware χρησιμοποιείται για το έλεγχο του τελικού κατασκευασμένου κυκλώματος και υπάρχει περίπτωση να συμβούν κατασκευαστικά λάθη σε ένα σωστό (επαληθευμένο) κύκλωμα.

4 Προσομοίωση

Υπάρχουν δύο κύρια είδη προσομοίωσης ψηφιακών κυκλωμάτων: λειτουργική (functional), και χρονική (timing) και διαφορετικοί τρόποι με τους οποίους εκτελούνται στο Quartus. Η functional προσομοίωση θεωρεί ότι τα καλώδια και οι λογικές πύλες έχουν μηδενική καθυστέρηση οπότε δεν είναι ρεαλιστική. Αλλά είναι πολύ γρήγορη και δίνει πληροφορία για το αν το κύκλωμα λειτουργεί σωστά. Η timing προσομοίωση παίρνει υπόψη της και τις καθυστερήσεις καλωδίων και πυλών, αλλά χρειάζεται η σχεδίαση να έχει προχωρήσει σε βάθος (π.χ. για να είναι γνωστά τα μήκη των καλωδίων) και είναι βέβαια πολύ πιο αργή. Έτσι η functional προσομοίωση είναι χρήσιμη στα αρχικά στάδια σχεδίασης για να επαληθεύουμε γρήγορα το κύκλωμά μας, πρίν ξοδέψουμε χρόνο για λεπτομερή σχεδίαση. Σε αυτό το μάθημα θα χρησιμοποιήσουμε μόνο functional προσομοίωση γιατί δεν θα κάνουμε λεπτομερή σχεδίαση, που είναι το αντικείμενο προαιρετικών μαθημάτων όπως η Ψηφιακή Σχεδίαση 2 και Κυκλώματα VLSI.

Για προσομοίωση θα χρησιμοποιήσουμε την εφαρμογή Modelsim Altera Starter Edition. Αν και μπορεί κανείς να τη ξεκινήσει από το Quartus, αυτό δεν δουλεύει καλά για τους σκοπούς τους μαθήματος. Επομένως θα τρέχουμε το Modelsim ανεξάρτητα.

Πριν ξεκινήσουμε τον προσομοιωτή, υπάρχουν 2 ακόμη δουλειές που πρέπει να γίνουν με το Quartus που περιγράφονται στα δύο επόμενα τμήματα.

4.1 Προετοιμασία

Ο προσομοιωτής δεν μπορεί να χειριστεί σχηματικά, οπότε η πρώτη δουλειά είναι να μετατρέψουμε τα σχηματικά σε Verilog. Για ευκολία έχω ετοιμάσει ένα απλό script που κάνει αυτή τη δουλειά, καλώντας μια εντολή του Quartus. Ονομάζεται `bdf2v` και βρίσκεται στον κατάλογο `~myy402/bin`. Μπορείτε να το αντιγράψετε αν έχετε δική σας εγκατάσταση του Quartus σε προσωπικό υπολογιστή. Το script υποθέτει ότι όλα τα αρχεία του Quartus project σας βρίσκονται στον ίδιο κατάλογο. Αν αυτό δεν ισχύει τότε θα πρέπει να κάνετε αλλαγές στο script ή στο τρόπο που οργανώνετε τα αρχεία του Quartus project σας.

Προσοχή αν αλλάξετε ένα σχηματικό, θα πρέπει να κάνετε ξανά τη μετατροπή πριν την προσομοίωση. Αλλιώς ο προσομοιωτής, που γνωρίζει μόνο τα αρχεία Verilog, θα χρησιμοποιεί την προηγούμενη έκδοση του κυκλώματος.

4.2 Test bench

Για να προσομοιώσουμε το κύκλωμα που σχεδιάσαμε πρέπει να του παρέχουμε τιμές εισόδων και, στη γενική περίπτωση, να παρατηρούμε και να ελέγχουμε αυτόματα την ορθότητα των εξόδων. Αυτές τις λειτουργίες τις παρέχει το λεγόμενο **testbench**. Πριν ξεκινήσουμε την προσομοίωση θα πρέπει να γράψουμε ένα testbench σε Verilog. Επειδή το κύκλωμα είναι απλό, το testbench δεν θα ελέγχει τις εξόδους, θα τις ελέγξουμε «με το μάτι».

Ο ευκολότερος τρόπος για να γίνει αυτό είναι χρησιμοποιώντας τον editor του Quartus με τον οποίο γράψαμε τον incrementer. Ανοίξτε λοιπόν έναν νέο αρχείο Verilog στο Quartus και δώστε του το όνομα `top.v` Το testbench θα πρέπει να περιέχει ένα instantiation του κυκλώματος υπό επαλήθευση που είναι το tutorial. Ένας τρόπος να βρείτε πως θα το κάνετε είναι να κοιτάξετε το αρχείο tutorial.v που δημιουργήθηκε από το script bdf2v. Στην αρχή του αρχείου θα δείτε τις δηλώσεις εισόδου εξόδου του module tutorial και από αυτές θα μπορείτε να γράψετε το instantiation.

Εναλλακτικά, στο Quartus και με ενεργό (επιλεγμένο) το tab του σχηματικού tutorial, επιλέγουμε File > Create/Update > Create Verilog Instantiation Template Files for Current File. Στη φόρμα που εμφανίζεται, παρατηρήστε το όνομα του αρχείου που θα δημιουργηθεί. Συνήθως είναι το όνομα του κυκλώματος ακολουθούμενο από το _inst.v.

Αν κοιτάξετε το αρχείο που δημιουργήθηκε (π.χ. more tutorial_inst.v), θα δείτε πως μπορείτε να κάνετε instantiate το κύκλωμα στο testbench. Για την ακρίβεια δείχνει έναν τρόπο instantiation που μπορεί να μην γνωρίζετε, αλλά είναι εξαιρετικός και σας τον προτείνω. Ονομάζεται instantiation with named ports όπου αντί η σειρά των εισόδων-εξόδων του module να ορίζει τις συνδέσεις, οι συνδέσεις γίνονται χρησιμοποιώντας τα ονόματα. Στην παρένθεση σε κάθε είσοδο έξοδο θα πρέπει να γράψει κανείς το όνομα του καλωδίου (reg ή wire) που χρησιμοποιείται στο testbench. Πιθανόν να θέλετε να αλλάξετε τα υπάρχοντα ονόματα μέσα στις παρενθέσεις με ονόματα που προτιμάτε.

Το κακό με αυτή τη μέθοδο είναι ότι δημιουργείται ένα ακόμη αρχείο και σε ένα μεγάλο project εύκολα μπερδεύεται κανείς. Θα πρότεινα αμέσως μετά την αντιγραφή του instantiation στο testbench, να διαγράφετε το αρχείο αυτό.

Συμπληρώστε τον υπόλοιπο κώδικα του testbench, χρησιμοποιώντας το παρακάτω ως υπόδειγμα. Θα το βρείτε ως αρχείο top.v στο lab05_starter στο GitHub.

```
module top;

reg        a, b, c, d, r;
wire [4:0] out;

tutorial duv(
    .a(a),
    .b(b),
    .c(c),
    .d(d),
    .r(r),
    .out(out)
);

initial begin
  a = 0;
  b = 0;
  c = 0;
  d = 0;
  r = 0;
  // out should be 5'h01
```

```
  #100
  a = 1;
  b = 0;
  c = 0;
  d = 0;
  r = 1;
  // out should be 5'h02
  #100
  a = 1;
  b = 1;
  c = 0;
  d = 0;
  r = 1;
  // out should be 5'h04
  #100 ; // propagate time, so last output is visible in waveforms.
end
endmodule
```

4.3 Προσομοίωση με Modelsim-Altera

Ξεκινήστε τον προσομοιωτή. Στους υπολογιστές των εργαστηρίων, στο ίδιο τερματικό όπου τρέχατε το Quartus[3], γράψτε `vsim &`. Σε Windows θα πρέπει να βρείτε που βρίσκεται το αντίστοιχο πρόγραμμα.

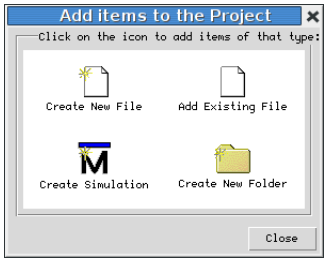
Για την προσομοίωση θα χρειαστούν 3 βήματα: δημιουργία ενός Modelsim project, compilation, και εκτέλεση της προσομοίωσης. Αν βρείτε λάθη, είτε στο κύκλωμα ή στο testbench, τα διορθώνετε και ξανακάνετε τα 2 τελευταία βήματα, μέχρι να μην υπάρχουν λάθη.

4.3.1 Δημιουργία ModelSim project

Από το μενού, επιλέξτε File > New > Project... Δώστε το όνομα tutorial_sim στο project, αφήστε τις υπόλοιπες επιλογές ως έχουν και πατήστε OK. Θα εμφανιστεί ένα μικρό παράθυρο σαν αυτό του σχήματος 3.

Αφού έχουμε ήδη δημιουργήσει τα αρχεία για το κύκλωμα και το testbench, επιλέξτε Add Existing File. Στο νέο παράθυρο που εμφανίζεται, πατήστε το Browse.. Διαλέξτε τα αρχεία top.v, tutorial.v, incrementer.v, κρατώντας το πλήκτρο Ctrl μαζί με κλικ για να διαλέξετε πολλαπλά αρχεία. Πατήστε OK για την επιλογή αρχείων και ξανά OK για το παράθυρο Add Existing File. Θα δείτε τα ονόματα των αρχείων να εμφανίζονται στο επάνω τμήμα του παραθύρου του ModelSim. Μπορείτε τώρα να κλείσετε το παράθυρο Add items to the Project.

[3]Χρειάζονται οι ρυθμίσεις του quartus.env.



Σχήμα 3: Το παράθυρο για το νέο ModelSim project

4.3.2 Compilation

Σε αυτό το βήμα ελέγχονται τα αρχεία Verilog για λάθη και δημιουργούνται κάποια προσωρινά δεδομένα που χρειάζονται για τη προσομοίωση. Από το μενού, επιλέξτε Compile > Compile All.

Αν δείτε λάθη, θα πρέπει να τα διορθώσετε στον Editor του Quartus, αν το πρόβλημα είναι σε αρχεία Verilog (όπως τα incrementer.v, top.v) ή αλλάζοντας το σχηματικό, για αρχεία που προέρχονται από σχηματικό. Στη δεύτερη περίπτωση θα πρέπει να ξανατρέξετε και το script bdf2v για να ανανεωθεί το αντίστοιχο αρχείο Verilog.

4.3.3 Προσομοίωση

Για να ξεκινήσετε την προσομοίωση, από το μενού επιλέξτε Simulate > Start Simulation... Θα εμφανιστεί ένα παράθυρο με πολλές «βιβλιοθήκες» (libraries). Βρείτε τη βιβλιοθήκη **work** που ήταν η προεπιλογή όταν είχατε δημιουργήσει το ModelSim project. Πατήστε το + δίπλα στη παραπάνω βιβλιοθήκη και θα δείτε τα περιεχόμενά της, τα 3 modules: incrementer, top, tutorial Επιλέξτε το top. Αυτό είναι το «υψηλότερο» verilog module που περικλείει τα υπόλοιπα, κάτι σαν την main στη Java. Πατήστε OK.

Θα δείτε ότι το παράθυρο του ModelSim αλλάζει μορφή. Πάνω αριστερά θα δείτε ένα παράθυρο που ονομάζεται Sim και στα δεξιά του άλλα δύο παράθυρα με ονόματα Objects και Processes.

Το παράθυρο Sim περιέχει την ιεραρχία των modules με τα ονόματα των instances. Συγκεκριμένα θα δείτε πρώτα το top, από κάτω του το duv, κλπ. Για τμήματα Verilog κώδικα που δεν είναι instances φαίνονται τα ονόματα που περιέχουν τις λέξεις ASSIGN, ALWAYS κλπ. Κάνοντας κλικ σε ένα instance εμφανίζονται, στο παράθυρο Objects, τα ονόματα των σημάτων (reg, wire, ...) που υπάρχουν μέσα σε αυτό.

Στην προσομοίωση θέλουμε να μπορούμε να βλέπουμε τις τιμές που έχουν διάφορα σήματα, είσοδοι, έξοδοι και εσωτερικοί κόμβοι, ώστε να μπορούμε να καταλάβουμε αν το κύκλωμα δουλεύει σωστά ή να βρούμε την αιτία του λάθους. Επειδή οι τιμές των σημάτων μεταβάλονται με το χρόνο, οι αναπαραστάσεις των τιμών τους λέγονται κυματομορφές, **waveforms**.

Πριν ξεκινήσει η προσομοίωση, πρέπει να πούμε στον προσομοιωτή, για ποιά σήματα να δημιουργήσει κυματομορφές. Αυτό γίνεται επιλέγοντας το όνομα του σήματος στο παράθυρο Objects και από το μενού Add > To Wave > Selected Signals, ή με δεξί κλικ και αντίστοιχες επιλογές. Μπορείτε να επιλέξετε πολλά σήματα μαζί πατώντας το Ctrl όταν κάνετε κλίκ στο όνομα σήματος. Αν θέλετε να προσθέσετε κυματομορφες από σήματα που βρίσκονται σε άλλο instance, προσθέστε τα σήματα του ενός, διαλέξτε το άλλο instance από το παράθυρο sim και μετά συνεχίστε τη διαδικασία όπως παραπάνω.

Για το κύκλωμα επιλέξτε τα σήματα a, b, c, d, r και out από το top και προσθέστε τα στο wave. Καν΄τε το ίδιο με το in από το instance duv. Θα δείτε ένα καινούριο παράθυρο, Wave να εμφανίζεται δίπλα στο Objects. Επειδή αυτό το παράθυρο θέλουμε να το αλλάζουμε θέση και μέγεθος ελεύθερα, πατήστε το κουμπί Undoc στη πάνω δεξιά μεριά του ώστε να γίνει ξεχωριστό παράθυρο από αυτό του ModelSim.

Το παράθυρο Wave έχει στα αριστερά τα ονόματα των σημάτων που διαλέξαμε, ένα χώρο στη μέση όπου θα εμφανίζονται οι τιμές των σημάτων και ένα μεγάλο μέρος δεξιά που θα εμφανιστούν οι κυματομορφές αλλά για την ώρα είναι άδειο.

Για να τρέξει η προσομοίωση βρείτε πάνω δεξιά τα εικονίδια που μοιάζουν με φύλλα χαρτιού με μερικές γραμμές πάνω τους και ένα βέλος δίπλα στο χαρτί. Πατήστε το εικονίδιο Run (ή πατήστε F9). θα δείτε ότι οι κυματομορφές αρχίζουν να εμφανίζονται. Πατήστε το μερικές φορές ακόμη μέχρι να σιγουρευτείτε ότι δεν οι κυματομορφές δεν αλλάζουν άλλο. Θα πρέπει να βλέπετε κάτι σαν το σχήμα 4.

Κάντε zoom στις κυματομορφές. Πολύ χρήσιμο είναι το Zoom Full ώστε να δείτε τις κυματομορφές στην πλήρη ανάπτυξή τους. Βεβαιωθείτε ότι το κύκλωμα δουλεύει σωστά: θα πρέπει το out να είναι πάντα μεγαλύτερο κατά 1 του in.

Αν είχατε ξεχάσει να προσθέσετε ένα σήμα στις κυματομορφές, μπορείτε να το κάνετε αργότερα, αλλά θα δείτε ότι δεν θα φαίνεται η κυματομορφή του για το προηγούμενο χρόνο προσομοίωσης, θα εμφανιστεί μόνο για τη συνέχεια της προσομοίωσης. Αν θέλετε να δείτε τη κυματομορφή από την αρχή,

πατήστε το εικονίδιο Restart και ξανατρέξτε την προσομοίωση από την αρχή.

Τέλος αν αλλάξετε είτε το κύκλωμα είτε το testbench, θα πρέπει να κάνετε compile, μετά Restart και να ξανατρέξετε την προσομοίωση.

Σε αυτό το σημείο έχετε ολοκληρώσει τη προσομοίωση. Δείτε το τμήμα Παραδοτέο για το τελικό βήμα που πρέπει να κάνετε μόνοι σας.

Μετά, αν έχετε χρόνο, εξερευνείστε το περιβάλλον του ModelSim. Βρείτε πως θα μπορείτε να δείτε τις τιμές των busses ως δεκαδικούς αριθμούς, πως θα μπορείτε να δείτε τις τιμές των bits ξεχωριστά, πώς να αλλάξετε τη σειρά που εμφανίζονται τα σήματα ώστε να σας βολεύει περισσότερο (π.χ. είσοδοι επάνω, έξοδοι κάτω). Υπαρχει επίσης η δυνατότητα να εκτελείτε γραμμή-γραμμή κώδικα Verilog και πολλά άλλα.

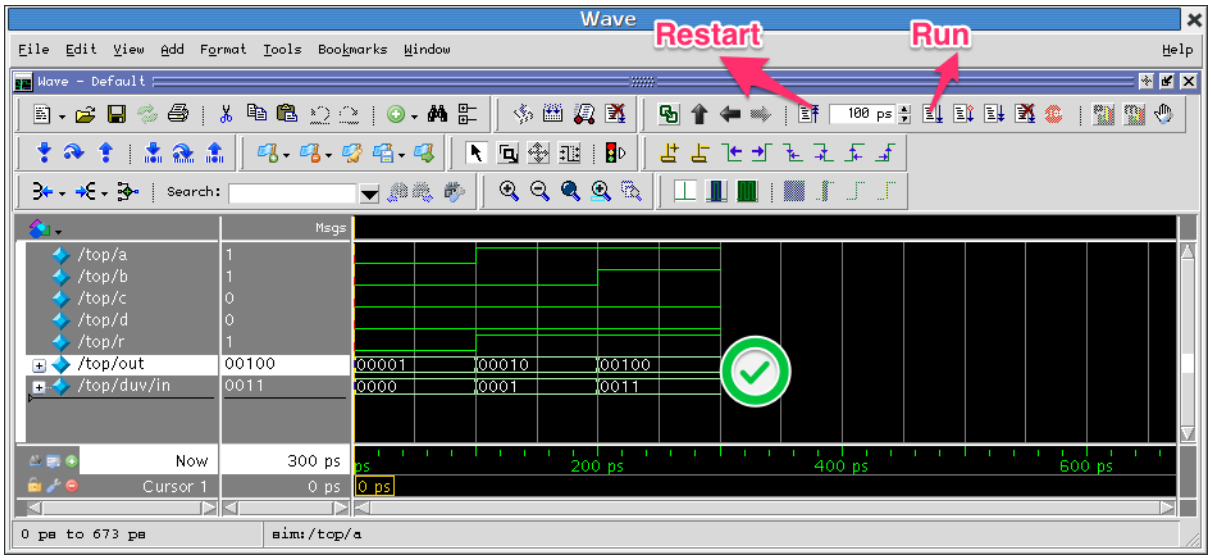
5 Παραδοτέο

Συμπληρώστε το testbench έτσι ώστε να έχει ως είσοδο τον αριθμό μητρώου σας modulo 16, δηλαδή διαιρέστε τον αριθμό μητρώου σας με το 16 και βάλτε το υπόλοιπο ως είσοδο στο κύκλωμα. Προσοχή το a είναι το λιγότερο σημαντικό bit και το d το περισσότερο σημαντικό. Το r χρησιμοποιείται ώς reset όταν είναι 0, οπότε αφήστε το σε 1.

Παραδόστε το αλλαγμένο testbench (top.v), το σχηματικό διάγραμμα (tutorial.bdf) και μια «φωτογραφία» (screenshot) του waveform που δείχνει την είσοδο (a-d, r) και έξοδο (out). Η φωτογραφία θα πρέπει να είναι σε ικανό μέγεθος για να μπορεί κανείς να δεί τους αριθμούς, περίπου στο μέγεθος του σχήματος 4. Η παράδοση θα γίνει μέσω GitHub σε ένα φάκελο/κατάλογο με όνομα lab05.

6 Καθαρισμός αρχείων

Στους υπολογιστές των εργαστηρίων, επειδή ο διαθέσιμος χώρος σας στο δίσκο είναι περιορισμένος (quota), και τα εργαλεία που χρησιμοποιήσατε δημιουργούν πολλά και μεγάλα αρχεία, όταν τελειώσετε με την άσκηση, σβήστε όλα τα περιτά αρχεία. Κρατήστε μόνο το σχηματικό διάγραμμα (tutorial.bdf), tutorial.v, incrementer.v, το σύμβολο του incrementer (incrementer.bsf), το project file (tutorial.qsf, tutorial.qpf), testbench (top.v), και το ModelSim project file (tutorial_sim.mpf)



Σχήμα 4: Το παράθυρο κυματομορφών.