

Ψηφιακά Ολοκληρωμένα Κυκλώματα VLSI-ASIC Μεγάλης Κλίμακας

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ

ΜΑΡΙΟΣ ΠΑΚΑΣ – ΔΙΠΛ. ΗΛ. ΜΗΧ. ΚΑΙ ΜΗΧ. ΥΠΟΛ.
ΑΡΙΣΤΟΤΕΛΗΣ ΤΣΕΚΟΥΡΑΣ – ΔΙΠΛ. ΗΛ. ΜΗΧ. ΚΑΙ ΜΗΧ. ΥΠΟΛ.
ΒΑΣΙΛΗΣ ΠΑΥΛΙΔΗΣ – ΑΝ. ΚΑΘΗΓΗΤΗΣ, ΤΗΜΜΥ
ΑΠΘ | ΕΡΓΑΣΤΗΡΙΟ ΗΛΕΚΤΡΟΝΙΚΗΣ

Περιεχόμενα

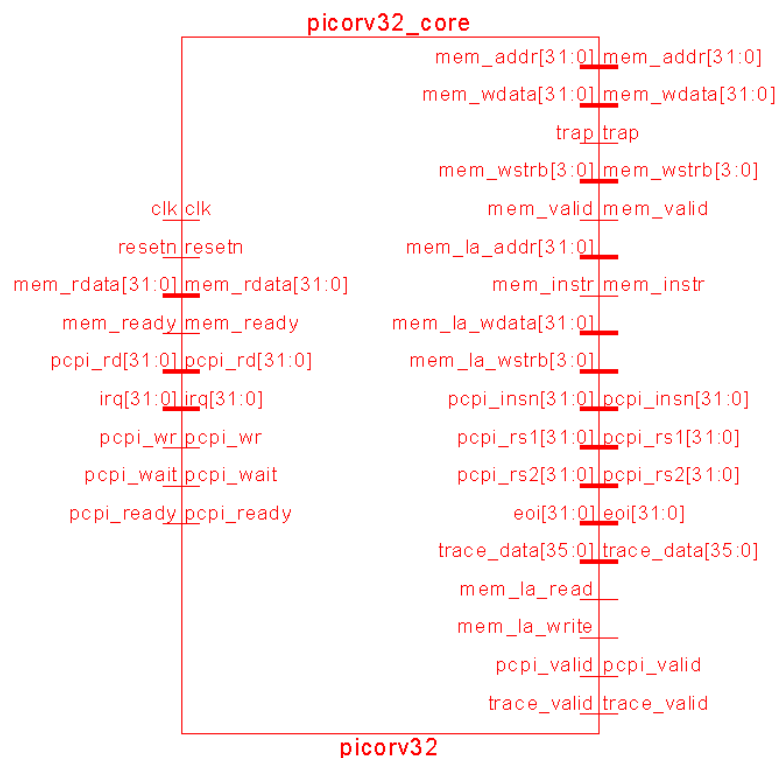
Εισαγωγή.....	2
Άσκηση 1.....	2
Άσκηση 2.....	7
Άσκηση 3.....	7
Άσκηση 4.....	7
Άσκηση 5.....	7
Άσκηση 6.....	7
Άσκηση 7.....	8
Άσκηση 8.....	8
Άσκηση 9.....	8
Παραδοτέα Ασκήσεων 1-9.....	11

Εισαγωγή

Το έγγραφο αυτό περιέχει υλικό σχετικά με τις ασκήσεις που καλείστε να ολοκληρώσετε στα πλαίσια του εργαστηρίου του μαθήματος. Θα πρέπει να χρησιμοποιηθεί σε συνδυασμό με το «Εγχειρίδιο του Εργαστηρίου». Σας προτείνετε να προχωρήσετε πέρα από την απλή ολοκλήρωση των ασκήσεων σε περισσότερη διερεύνηση των χαρακτηριστικών των εργαλείων ή/και να δοκιμάσετε τη ροή σχεδίασης σε κάποιο άλλο κύκλωμα που επιθυμείτε.

Κάθε κύκλωμα που σχεδιάζετε, βασίζεται σε ένα σύνολο βιβλιοθηκών που περιγράφουν διάφορα χαρακτηριστικά των κελιών και των τρανζίστορ που τα αποτελούν. Οι βιβλιοθήκες αυτές είναι συνάρτηση της τεχνολογίας καρασκευής που έχει επιλεγεί για το κύκλωμα αυτό. Για το μάθημα αυτό, έχει επιλεγεί η βιβλιοθήκη της διεργασίας κατασκευής (**GPDK45** (Generic Process Design Kit 45)) που υποδεικνύει μία τεχνολογία κατασκευής (technology node ή process) στα 45 nm. Αν και η τεχνολογία αυτή δεν προέρχεται από κάποια συγκεκριμένη βιομηχανία κατασκευής ολοκληρωμένων κυκλωμάτων (μας απαλλάσσει από τη διαδικασία υπογραφής άδειας χρήσης της τεχνολογία για καθέναν από εσάς!), μας επιτρέπει να ακολουθήσουμε όλα τα βήματα της ροής σχεδίασης ASIC κυκλωμάτων. Αποτελείται από δύο βιβλιοθήκες, μία για τα κελιά (standard cells) και μία για τις εισόδους/εξόδους του κυκλώματος, που ονοματίζονται ως **gsclib45** και **giolib45**, αντίστοιχα.

Το κύκλωμα που σας δίνετε για να εργασθείτε σε αυτό είναι ένα κύκλωμα σε γλώσσα Verilog και υλοποιεί έναν πυρήνα επεξεργαστή αρχιτεκτονικής RISC-V. Μια υλοποίηση σε υλικό του αλγορίθμου αυτού φαίνεται στο Σχήμα 1. Το κύκλωμα αυτό περιγράφεται ολόκληρο στο αρχείο Verilog picorv32.v.



Σχήμα 1. Υλοποίηση σε υλικό του αλγορίθμου AES.

Άσκηση 1

Σκοπός της άσκησης αυτής είναι να κατανοήσετε τη διαδικασία της σύνθεσης και φυσικής σχεδίασης ψηφιακών κυκλωμάτων, και να την εφαρμόσετε σε ένα απλό και επίπεδο (flat) κύκλωμα. Τα βήματα

της διαδικασίας σύνθεσης ενός κυκλώματος με το εργαλείο genus που χρησιμοποιούμε στο μάθημα αυτό εικονίζονται στο Σχήμα 2. Σας δίνεται λοιπόν ένα απλό κύκλωμα και καλείστε, εφόσον κατανοήσετε τη λειτουργία του, (για το σκοπό αυτό μπορείτε να επισκεφτείτε το repository στο github και να διαβάσετε την περιγραφή του επεξεργαστή: <https://github.com/YosysHQ/picorv32>, καθώς και να τον τεστάρτε με χρήση κατάλληλων toolchains) να προχωρήσετε στη σύνθεση του κυκλώματος και έπειτα στη φυσική σχεδίαση του με τη διαδικασία που περιγράφεται στο Εγχειρίδιο του Εργαστηρίου.

Βήμα 1: Για τη διαδικασία της σύνθεσης, θα πρέπει να ορίσετε τα μονοπάτια για τις βιβλιοθήκες χρονισμού (*.lib), φυσικών πληροφοριών (*.lef) και παρασιτικών (.tch). Οι βιβλιοθήκες αυτές βρίσκονται στο μονοπάτι

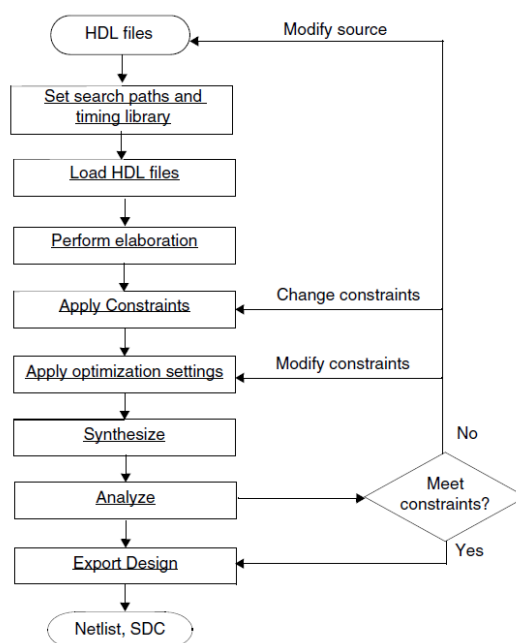
```
/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gscli  
b045_svt_v4.4/gsclib045
```

στο φάκελο **timing**, στο φάκελο **lef** και στο φάκελο **qrc**, αντίστοιχα.

Σημείωση: Αφιερώστε λίγο χρόνο να καταλάβετε κάποια χαρακτηριστικά των διαθέσιμων βιβλιοθηκών και το τί σημαίνουν τα ονόματά τους επίσης.

Στον φάκελο timing επιλέξτε μία βιβλιοθήκη καθυστέρησης που θέλετε να χρησιμοποιήσετε για τη σχεδίαση του κυκλώματος. Στον φάκελο lef πρέπει να επιλέξετε τα δύο αρχεία tech και macro, τα οποία περιέχουν φυσικές πληροφορίες σχετικά με τα επίπεδα μετάλλων και τα κελιά, αντίστοιχα. Στον φάκελο qrc επιλέξτε το αρχείο παρασιτικών (.tch) και εισάγετέ το στο inponus μαζί με τα υπόλοιπα αρχεία όταν φτάσετε στο κατάλληλο βήμα των ασκήσεων. Επίσης, πρέπει να θέσετε τα μονοπάτια όπου βρίσκονται τα αρχεία Verilog του κυκλώματος, καθώς και τα μονοπάτια από τα οποία μπορείτε να καλέσετε διάφορα scripts (π.χ. αρχεία περιορισμών sdc) για να καθοδηγήσετε τη διαδικασία της σύνθεσης.

Προσοχή: Να χρησιμοποιηθεί η βιβλιοθήκη χρονισμού με όνομα fast_vdd1v0_basicCells.lib.



Σχήμα 2. Βήματα της βασικής διαδικασίας σύνθεσης με το εργαλείο genus®.

Αφού θέσετε τα μονοπάτια αυτά, πρέπει να διαβάσετε τις συγκεκριμένες βιβλιοθήκες. Προσέξτε τα μηνύματα που τυχόν παράγει το εργαλείο κατά τη διάρκεια ανάγνωσής τους. Τί παρατηρείτε;

Βήμα 2: Έχοντας κάνει τις αρχικές ρυθμίσεις, στο βήμα αυτό διαβάζουμε τα αρχεία που περιγράφουν το κύκλωμα μας.

Ερώτηση: Ποιό είναι το top-level module του κυκλώματός σας; Με ποιά εντολή μπορώ να το προσδιορίσω αυτό; **Απάντηση:** _____

Βήμα 3: Αν έχετε υλοποιήσει τα προηγούμενα βήματα σωστά, στο βήμα αυτό γίνεται μία πρώτη ανάλυση του κυκλώματος (elaboration). Μετά την επιτυχή εκτέλεση του βήματος αυτού, προτείνεται ο έλεγχος του κυκλώματος/σχεδίασης για τυχόν προβλήματα (check_design). Ελέγξτε αν υπάρχουν τυχόν «unresolved references» και προσπαθήστε να τις επιλύσετε αν τυχόν υπάρχουν.

Βήμα 4: Πριν προχωρήσουμε στη σύνθεση του κυκλώματος πρέπει να θέσουμε διάφορους περιορισμούς που θα καθοδηγήσουν τη διαδικασία αυτή. Το βήμα αυτό είναι ίσως το πιο σημαντικό βήμα για το αποτέλεσμα της σύνθεσης. Για το σκοπό αυτό πρέπει να δημιουργήσετε ένα αρχείο περιορισμών *.sdc, το οποίο και θα αποθηκεύσετε στο μονοπάτι των scripts που δημιουργήσατε στο Βήμα 1. Δε σας δίνονται οι εντολές αυτές αλλά σας παρέχονται παρακάτω οδηγίες για το είδος των περιορισμών που απαιτούνται (δεν είναι μοναδικοί, αλλά σας δίνετε ένας βασικός αριθμός περιορισμών επαρκής για να προχωρήσετε στη σύνθεση του κυκλώματος). Έχετε απόλυτη ελευθερία να αλλάξετε, προσθέσετε, αφαιρέσετε περιορισμούς. Σε αυτήν την περίπτωση θα πρέπει να εξηγήσετε στην αναφορά σας (πολύ σύντομα), τι στόχο έχουν οι αλλαγές ή προσθήκες αυτές. Επίσης, προτείνεται πριν από κάθε περιορισμό να προσφέρεται ένα σύντομο σχόλιο (θυμηθείτε ότι γραμμές σχολίων ξεκινούν με το χαρακτήρα '#').

Επιθυμητοί περιορισμοί για τη σύνθεση:

- 1) Ορίστε ένα ρολόι με 50% duty cycle, περίοδο 10 ns, και όνομα clk
- 2) Ορίστε την καθυστέρηση (latency) του σήματος αυτού του ρολογιού στα 400 ps
- 3) Ορίστε την αβεβαιότητα του σήματος ρολογιού στα 50 ps *0.5 ns*
- 4) Ορίστε το χρόνο μετάβασης (ανόδου/καθόδου) του ρολογιού στο 1% της περιόδου του ρολογιού
- 5) Ορίστε την καθυστέρηση των εξόδων (output delay) για όλες τις εξόδους του κυκλώματος για την ανάλυση χρονισμού setup *1 ns* όπου θα συμπεριλαμβάνεται και η καθυστέρηση του δικτύου διανομής του σήματος ρολογιού (clock network)
- 6) Ορίστε την καθυστέρηση των εξόδων (outputs) για όλες τις εξόδους του κυκλώματος για την ανάλυση χρονισμού hold *0.4 ns* όπου θα συμπεριλαμβάνεται και η καθυστέρηση του δικτύου διανομής του σήματος ρολογιού (clock network)
- 7) Ορίστε το φορτίο όλων των εξόδων για ανάλυση χρονισμού setup στα 0.5 pF
- 8) Ορίστε το φορτίο όλων των εξόδων για ανάλυση χρονισμού hold στα 0.01 pF
- 9) Ορίστε την καθυστέρηση των εισόδων (input delay) για όλες τις εισόδους του κυκλώματος για την ανάλυση χρονισμού setup στο 1 ns όπου θα συμπεριλαμβάνεται και η καθυστέρηση του δικτύου διανομής του σήματος ρολογιού (clock network)
- 10) Ορίστε την καθυστέρηση των εισόδων (input delay) για όλες τις εισόδους του κυκλώματος για την ανάλυση χρονισμού hold στο 0.4 ns όπου θα συμπεριλαμβάνεται και η καθυστέρηση του δικτύου διανομής του σήματος ρολογιού (clock network)

11) Ορίστε ένα κελί από τη βιβλιοθήκη που επιλέξατε στο Βήμα 1 για την οδήγηση όλων των εισόδων του κυκλώματος. Προτείνεται η χρήση του κελιού BUFX2 για την ανάλυση setup και η χρήση του κελιού BUFX16 για την ανάλυση hold.

Σημείωση: Οι μονάδες χρόνου και χωρητικότητας των αρχείων .sdc είναι ανάλογες με αυτές στο αρχείο χρονισμού (.lib) που χρησιμοποιήσαμε (για την περίπτωση μας αυτές είναι nanosecond (ns) και picofarad (pF), αντίστοιχα). Το genus θα μετατρέψει τις μονάδες αυτές σε picosecond (ps) και femtofarad (fF), αντίστοιχα.

Αφού διαβάσετε το αρχείο περιορισμών και δεν αναφέρονται κάποια λάθη, προτείνεται ο έλεγχος των περιορισμών με τη χρήση της εντολής `check_timing_intent`. Δείτε τις σελίδες βοήθειας για τη χρήση και σκοπό της εντολής αυτής.

Προσοχή: Το εργαλείο σύνθεσης μερικές φορές χρησιμοποιεί σκαν φλιπ φλοπ (SDFF*) αντί για κανονικά φλιπ φλοπ (DFF*) χωρίς να υπάρχουν αλυσίδες ανίχνευσης ορισμένες από τον χρήστη. Αυτό δημιουργεί πρόβλημα κατά την τοποθέτηση με το innovus (το εργαλείο ζητάει να οριστούν αλυσίδες παρόλο που δεν υπάρχουν). Για να αποφευχθεί το συγκεκριμένο πρόβλημα, **εφόσον δεν έχετε σκοπό να ορίσετε αλυσίδες ανίχνευσης στη σύνθεση**, χρησιμοποιήστε την εντολή:

```
set_db / .use_scan_seqs_for_non_dft false
```

Βήμα 5: Αν δεν υπάρχουν λάθη στους περιορισμούς που θέσατε, είστε πλέον έτοιμοι να συνθέσετε το κύκλωμα σας. Εκτελέστε και τα τρία βήματα σύνθεσης (generic, mapping, optimization).

Βήμα 6: Έχοντας συνθέσει επιτυχώς το κύκλωμα σας, μπορείτε να αποτιμήσετε τα χαρακτηριστικά του κυκλώματος σας. Για το σκοπό αυτό θα πρέπει να συμπεριλάβετε στην αναφορά σας τα ακόλουθα: αριθμός κελιών, επιφάνεια, slack, και ισχύ του κυκλώματος. Τα στοιχεία αυτά θα πρέπει να παρουσιαστούν με τη βοήθεια ενός πίνακα στην αναφορά σας. Επίσης, μπορείτε να δείτε μία περίληψη των αποτελεσμάτων της σύνθεσης και της ποιότητας των αποτελεσμάτων (quality of results (qor)). Μπορείτε να παράγετε τα σχετικά αρχεία μέσω των εντολών που σας δίνονται στο εγχειρίδιο του εργαστηρίου.

Βήμα 7: Παράγετε όλα τα απαραίτητα αρχεία για τα επόμενα στάδια της ροής σχεδίασης όπου θα χρησιμοποιηθεί το εργαλείο innovus. Αυτό μπορεί να γίνει με την εντολή `write_design -innovus <top_level_module>`, η οποία θα δημιουργήσει έναν φάκελο με το όνομα `genus_invs_des` που θα περιέχει όλα τα απαραίτητα αρχεία.

Βήμα 8: Εισάγετε τα αρχεία στο innovus και αρχικοποιήστε το design. Για αυτό το βήμα θα χρησιμοποιήσετε τα αρχεία `netlist`, `sdc`, `mmmc.tcl` που εξάχθηκαν από το genus, καθώς και τα `.lef`, `.lib` και `.tch` αρχεία που χρησιμοποιήσατε στα βήματα 1-6 για τη σύνθεση.

Βήμα 9: Εφαρμόστε τη χωροθέτηση στο design με ποσοστό χρήσης του πυρήνα **75%**. Επίσης, για το δίκτυο διανομής ισχύος, δημιουργήστε χώρο 15 μm μέχρι τα I/O για τους δακτύλιους.

Βήμα 10: Δημιουργήστε το δίκτυο διανομής ισχύος, με τους δακτύλιους να χρησιμοποιούν τα δύο ανώτερα μέταλλα. Οι δακτύλιοι πρέπει να έχουν πάχος 3 μm, κενό διάστημα ανάμεσα τους επίσης 3 μm, και να βρίσκονται κεντραρισμένοι στο διάκενο ανάμεσα στον πυρήνα και στα I/O. Όσον αφορά τις γραμμές, επιλέξτε παρόμοιο πάχος και κενό, και αριθμό σελίδων ίσο με τρία. Επίσης, δημιουργήστε τους ακροδέκτες I/O τροφοδοσίας και γείωσης κατάλληλα, έτσι ώστε να συνδέονται με τους

δακτύλιους γύρω από τον πυρήνα. Τέλος, δημιουργήστε τα follow pins για τη σύνδεση των κελιών με το δίκτυο διανομής ισχύος.

Βήμα 11: Να γίνει Early Power Rail Ανάλυση και να αποτιμηθεί η ικανότητα του δικτύου διανομής να τροφοδοτήσει κάθε κόμβο του κυκλώματος. Το συγκεκριμένο βήμα να ακολουθηθεί μόνο για την Άσκηση 1, ενώ για τις υπόλοιπες είναι στην ευχέρεια σας αν θέλετε να το επαναλάβετε.

Βήμα 12: Ρυθμίστε τη βελτιστοποίηση, επιλέγοντας **high effort** για το χρονισμό και **χωρίς** βελτιστοποίηση της ισχύος. Ύστερα, εκτελέστε την τοποθέτηση. Μετά το πέρας της, εξάγεται αναφορές για τα slack, την κατανάλωση ισχύος ως προς τον κάθε τύπο ισχύος και για την επιφάνεια. Τα στοιχεία αυτά θα πρέπει επίσης να παρουσιασθούν με τη βοήθεια ενός πίνακα.

Βήμα 13: Εφαρμόστε Early Global Routing με εύρος δρομολόγησης που να καλύπτει όλα τα μέταλλα (δηλαδή επιτρέψτε στο δρομολογητή να χρησιμοποιήσει όλα τα επίπεδα μετάλλου για τη διασύνδεση των κελιών του κυκλώματος). Δοκιμάστε δεύτερη φορά με εύρος από το δεύτερο (M2) έως το προ-προτελευταίο μέταλλο (M10). Υπάρχει συμφόρηση σε κάποια από τις δύο περιπτώσεις στο design;

Βήμα 14: Εκτελέστε τη σύνθεση του δέντρου ρολογιού. Για αυτή την περίπτωση δημιουργήστε ένα NDR (Non Default Rule) με διπλάσιο πάχος και διπλάσιο κενό για όλα τα μέταλλα. Ύστερα δημιουργήστε έναν τύπο δρομολόγησης που να ακολουθεί το συγκεκριμένο NDR και έχει μέγιστο επίπεδο μετάλλου, δηλαδή το **αμέσως κατώτερο από το ελάχιστο επίπεδο που χρησιμοποιεί το δίκτυο διανομής της ισχύος**. Επιλέξτε το ελάχιστο έτσι ώστε το **εύρος μεταξύ ελάχιστου και μέγιστου επιπέδου μετάλλου για το δέντρο ρολογιού να περιέχει τρία επίπεδα**. Επίσης, προσθέστε επιθυμητή στρέβλωση 100 ps και μέγιστο ρυθμό μετάβασης του ρολογιού ίσο με 150 ps. Μετά το πέρας της σύνθεσης του δέντρου ρολογιού, εξάγετε τις αναφορές για τα slack, την κατανάλωση ισχύος και την επιφάνεια αφού κάνετε βελτιστοποίηση του κυκλώματος (design), και να τα αναφέρετε σε έναν πίνακα όπως και στο βήμα 12. Απαντήστε σε ένα χωριστό πίνακα, τις εξής ερωτήσεις:

- Πόσοι buffers ρολογιού χρησιμοποιούνται;
- Πόσα skew groups υπάρχουν στο design;
- Ικανοποιούνται ο στόχος για τη στρέβλωση και το μέγιστο ρυθμό μετάβασης που θέσαμε; (Ναι/όχι και γιατί;)
- Ποιό είναι το μέγιστο και το ελάχιστο βάθος του δέντρου ρολογιού;
- Ποιό είναι το μήκος δρομολόγησης για το trunk και ποιο για τα leaves ;

Βήμα 15: Εκτελέστε τη δρομολόγηση, επιλέγοντας τις ρυθμίσεις **Fix Antenna, SI Driven** και **Timing Driven** με **effort 5**, καθώς και **Medium Effort** για το **Via Optimization**, εξάγετε μετά τη βελτιστοποίηση τις αναφορές σχετικά με τα slack, την επιφάνεια και την κατανάλωση της ισχύος, και αναφέρετε τα αποτελέσματα αυτά σε έναν πίνακα (όπως αυτό στο βήμα 12).

Βήμα 16: Εξετάστε το φυσικό σχέδιο για DRC παραβιάσεις και προσπαθήστε να τις επιλύσετε εάν υπάρχουν. Ελέγξτε αν όλες οι συνδέσεις και οι τοποθετήσεις των κελιών έχουν γίνει ορθά. Τέλος, εισάγετε τα Fillers σε όλα τα επίπεδα για να ικανοποιήσετε την πυκνότητα των μετάλλων (**Σημαντικό:** Θέστε το ελάχιστο όριο της πυκνότητας ως **10%** για όλα τα μέταλλα). Οι υπόλοιπες ρυθμίσεις να παραμείνουν ίδιες.

Άσκηση 2

Εκτελέστε τα βήματα 12-15 της Άσκησης 1 (με τα ίδια αρχεία της σύνθεσης και το ίδιο δίκτυο διανομής της ισχύος που έχετε παράξει) επιλέγοντας αυτήν τη φορά βελτιστοποίηση της δυναμικής ισχύος στο βήμα 12 με **high effort**. Αναφέρετε τί διαφορές υπάρχουν μετά την εκτέλεση των βημάτων 12, 14, και 15 στην ισχύ, επίδοση, και επιφάνειας συγκριτικά με τα αντίστοιχα βήματα της Άσκησης 1 και που μπορεί να οφείλονται; (Παρέχετε ένα πίνακα παρόμοιο με της Άσκησης 1 για τα βήματα αυτά συνολικά και μία σύντομη συζήτηση γιατί παρατηρούνται οι διαφορές αυτές (1 παρ.)).

Άσκηση 3

Εκτελέστε τα βήματα 9-15 της Άσκησης 1 (με τα ίδια αρχεία της σύνθεσης) επιλέγοντας ποσοστό χρήσης του πυρήνα **85%** στο Βήμα 9. Αναφέρετε τί διαφορές υπάρχουν μετά την εκτέλεση των βημάτων 12, 14, και 15 στην επιφάνεια συγκριτικά με τα αντίστοιχα βήματα της Άσκησης 1 και που μπορεί να οφείλονται; (Παρέχετε ένα πίνακα παρόμοιο με της Άσκησης 1 για τα βήματα αυτά συνολικά και μία σύντομη συζήτηση γιατί παρατηρούνται οι διαφορές αυτές (1 παρ.)). Τί διαφορά φαίνεται να υπάρχει στο μήκος των διασυνδέσεων και τον αριθμό των νία σε κάθε περίπτωση; (εξηγήστε σε 1 παράγραφο γιατί απαντάται αυτή η διαφορά).

Άσκηση 4

Επαναλάβετε τα Βήματα 1-16 για μία συχνότητα ρολογιού **8 ns**. Παράγετε τον πίνακα των αποτελεσμάτων της σύνθεσης που περιγράφεται στο Βήμα 6 (δεν είναι απαραίτητη η εξαγωγή περίληψης και ποιότητας αποτελεσμάτων από το genus για την άσκηση αυτή), καθώς και έναν πίνακα με τα αποτελέσματα (ισχύος, επιφάνειας, επίδοσης) για τα στάδια PreCTS, PostCTS, PostRoute, από τα βήματα 12, 14, και 15 αντίστοιχα. Τί διαφορές υπάρχουν σε σχέση με τα αποτελέσματα της Άσκησης 1 και γιατί (1 παράγραφος);

Άσκηση 5

Επαναλάβετε τα Βήματα 1-16 της Άσκησης 1 για τη βιβλιοθήκη χρονισμού με όνομα **slow_vdd1v0_basicCells.lib** από αυτές που είναι διαθέσιμες στο μονοπάτι των βιβλιοθηκών, και παράγετε τον πίνακα των αποτελεσμάτων της σύνθεσης που περιγράφεται στο Βήμα 6 για την καινούργια βιβλιοθήκη (δεν είναι απαραίτητη η εξαγωγή περίληψης και ποιότητας αποτελεσμάτων από το genus για την άσκηση αυτή), καθώς και έναν πίνακα με τα αποτελέσματα (ισχύος, επιφάνειας, επίδοσης) για τα στάδια PreCTS, PostCTS, PostRoute, από τα βήματα 12, 14, και 15. Τί διαφορές υπάρχουν σε σχέση με τα αποτελέσματα της Άσκησης 1 και γιατί (1 παράγραφος);

Άσκηση 6

Επαναλάβετε τα Βήματα 1-16 της Άσκησης 1 όπου θα πρέπει να ενεργοποιήσετε τη φραγή του ρολογιού στα αρχικά βήματα της σύνθεσης. Καλείστε (α) να εξάγετε τα στατιστικά της χρήσης φραγής του ρολογιού, παράγετε τον πίνακα των αποτελεσμάτων της σύνθεσης που περιγράφεται στο Βήμα 6 και να τα συγκρίνετε με εκείνα της Άσκησης 1 και (β) να εξάγετε τα αποτελέσματα (ισχύος, επιφάνειας, επίδοσης) για τα στάδια PreCTS, PostCTS, PostRoute από τα βήματα 12, 14, και 15 αντίστοιχα και να σημειώσετε τις διαφορές και τις παρατηρήσεις σας (1 παράγραφος). Υπόδειξη: Για την εξαγωγή των στατιστικών χρήσης, ενδείκνυται η εντολή `report_clock_gating`, που αναφέρεται και στο εγχειρίδιο χρήσης.

Άσκηση 7

Επαναλάβετε τα βήματα 1-7 της Άσκησης 1, δηλαδή ακολουθήστε τη διαδικασία της σύνθεσης μόνο. Αυτή τη φορά είναι σημαντικό να αποθηκεύσετε τα στιγμιότυπα μετά (i) το βήμα του elaboration, (ii) της generic, και (iii) της mapped σύνθεσης, προκειμένου να χρησιμοποιηθούν ως είσοδοι για το Conformal Equivalence Checking. Στη συνέχεια χρησιμοποιήστε το Conformal Logic Equivalence Checking πρόγραμμα της Cadence για να συγκρίνετε τα στιγμιότυπά σας. Αρχικά συγκρίνετε τον κώδικα RTL με το στιγμιότυπο που αποθηκεύσατε μετά το βήμα elaborate. Διαβάστε το output και κρατήστε την έξοδο της εντολής `report_verification -hier -verbose`. Επαναλάβετε την ίδια σύγκριση, αυτή τη φορά ανάμεσα στα στιγμιότυπα `syn_generic` και `syn_map` και κρατήστε την έξοδο της εντολής `report_verification -verbose`. Παραθέστε στην αναφορά σας τα αποτελέσματα των δύο αυτών συγκρίσεων και σχολιάστε αν ο κώδικας παρέμεινε λειτουργικά ίδιος ή όχι (1 παράγραφος).

ΤΡΕΧΕΙ ΣΤΟΝ ΑΡΙΘ. 8, ΔΟΚΙΜΑΝΕ ΝΑ ΔΕΙΧΝΕΙ ΥΠΕΡ ΤΟ ΕΞΕΛΕΞΑΙΟ

Να επαναληφθούν τα Βήματα 8-16 της Άσκησης 1, αλλά αυτή τη φορά μετά το βήμα της σύνθεσης να τροποποιηθεί ο κώδικας που έχει πλέον συντεθεί και να προστεθούν pads. Θα πρέπει επίσης κατά την εισαγωγή του project στο Innovus να περιληφθεί το αρχείο .lef που περιλαμβάνει την περιγραφή των pads που θα χρησιμοποιήσετε. Για την προσθήκη των pads χρησιμοποιήστε το script σε python που σας δίνεται. Ζητείται ένα στιγμιότυπο οθόνης (screenshot) με το κύκλωμα μετά τη δρομολόγηση, καθώς επίσης και αναφορά στην επιφάνεια ολόκληρου του κυκλώματος με τα pads, και ξεχωριστά την επιφάνεια που καταλαμβάνει ο πυρήνας. Τι παρατηρείτε;

ΝΕΟ PROJECT 5

Άσκηση 9

Τα ολοκληρωμένα κυκλώματα μεγάλης κλίμακας ολοκλήρωσης (VLSI) ελέγχονται ως προς τη σωστή λειτουργία τους μετά την κατασκευή τους. Ο έλεγχος αυτός πραγματοποιείται με (πολλαπλά) διανύσματα δοκιμών που παράγονται μέσω ειδικού εξοπλισμού ελέγχου (Automatic Test Equipment (ATE)). Για τις δοκιμές αυτές απαιτείται επιπλέον λογική που καθιστά το κύκλωμα κατάλληλο για δοκιμή (test). Επομένως, κατά τη διαδικασία της σύνθεσης πρέπει να φροντίσουμε ώστε η λογική αυτή να προστεθεί στο κύκλωμα, χωρίς όμως φυσικά να μεταβάλλουμε καθόλου την αρχική λειτουργία του κυκλώματος. Το βήμα αυτό στη ροή σχεδίασης ονομάζεται εν γένει «Design For Testability». Μετά την πρόσθεση της επιπλέον λογικής για DFT, πρέπει να διασφαλίζεται ότι το κύκλωμα από άποψη λογικής λειτουργίας παραμένει ακριβώς το ίδιο με το αρχικό κύκλωμα. Για το σκοπό αυτό υπάρχουν εργαλεία που επιβεβαιώνουν την (μη) ισοδυναμία των δύο κυκλωμάτων, και αναφέρουν στο σχεδιαστή πού και τί έχει αλλάξει αν διαπιστωθεί μη ισοδυναμία. Τα εργαλεία αυτά και γενικότερα το βήμα αυτό, ονομάζεται «Logic Equivalence Checker (LEC)» και *logic equivalence checking*, αντίστοιχα. Στην άσκηση αυτή καλείστε να εξοικειωθείτε με το βήμα που μετατρέπει ένα κύκλωμα σε «ικανό προς δοκιμή» κύκλωμα (DFT).

Επομένως, για το κύκλωμα που έχουμε δει μέχρι τώρα (Άσκηση 1), εκτελέστε τις παρακάτω εντολές (προσαρμοσμένες στο κύκλωμά σας) και να εξάγετε εκ νέου τα αποτελέσματα που σας ζητήθηκαν για τις προηγούμενες καταστάσεις, δηλαδή τον πίνακα με τα διάφορα χαρακτηριστικά του κυκλώματος μετά τη σύνθεση. Οι παρακάτω εντολές πρέπει να εφαρμοσθούν μετά το βήμα elaborate και πριν το βήμα της σύνθεσης (syn_generic). Τα σχόλια της χρήσης των εντολών δίνονται κυρίως στα Αγγλικά (καθώς κρίνεται ότι η μετάφραση μάλλον θα μείωνε την ακρίβεια περιγραφής αυτών).

Εντολές για εισαγωγή λογικής που προσφέρει DFT σε ένα κύκλωμα προς σύνθεση:

```
set_db / .dft_scan_style muxed_scan
```

- Η ιδιότητα αυτή προσδιορίζει τον τρόπο δοκιμής για όλα τα modules του κυκλώματος που διαβάζονται κατά το βήμα (read_hdl). Υπάρχουν οι ακόλουθοι τρόποι δοκιμής: muxed_scan και clocked_lssd_scan. Ο πιο δημοφιλής τρόπος είναι ο muxed_scan. Σημειώστε ότι ο χαρακτήρας slash ('/') σημαίνει ότι η ιδιότητα αυτή εφαρμόζεται στο root της ιεραρχίας σχεδίασης του εργαλείου genus ενώ το σημείο στίξης της τελείας ('.') πριν την ιδιότητα, υποδηλώνει ότι αυτό που ακολουθεί είναι ιδιότητα (attribute).

```
set_db / .dft_prefix DFT_
```

- This attribute specifies the prefix for instances of added control logic, and the base names for any scan-data input, scan-data output, shift-enable, scan_clock_a and scan_clock_b ports, or ports for compression test signals created during test synthesis.

```
set_db / .dft_identify_top_level_test_clocks true
```

- This attribute indicates whether the DFT rule checker can automatically assign a "test clock" for each top-level clock pin that is traceable from the clock pin of a flip-flop, and a corresponding "test-clock domain". To prevent auto identification, set this attribute to false.

```
set_db / .dft_identify_test_signals true
```

- This attribute indicates whether the DFT rule checker can automatically assign a test mode signal for each top-level pin that is traceable from the async set or reset pin of a flip-flop. To prevent auto identification, set this attribute to false.

```
set_db / .dft_identify_internal_test_clocks false
```

- This attribute indicates whether the DFT rule checker must identify the output pins of multi-input combinational gates and the clock output pins of the clock gating instances in the clock path as separate test clocks in the same DFT clock domain as its root-level test clock.

```
set_db / .use_scan_seqs_for_non_dft false
```

- During synthesis, scan mapping is bypassed. To prevent inferred flip-flops from being mapped to scan flip-flops for functional use, set the use_scan_seqs_for_non_dft root attribute to false before you run initial synthesis on the design.

```
set_db "design:<design_name>" .dft_scan_map_mode tdrc_pass
```

- This attribute controls the mapping of the flip-flops to their scan-equivalent flipflops.

```
set_db "design:<design_name>"\
```

```
.dft_connect_shift_enable_during_mapping tie_off
```

- This attribute controls the connection of the following pins during an initial synthesis run (syn_map), when mapping generic flops to their scan-equivalent flops.

```
set_db "design:<design_name>"\
```

```
.dft_connect_scan_data_pins_during_mapping loopback
```

- Starting from RTL, this attribute controls the connections of the scan-data pins during an initial synthesis run (syn_map), when mapping generic flops to their scan-equivalent flops. Starting from a gate-level netlist, this attribute controls the connections of the scan-data pins during an incremental synthesis run (syn_opt), when remapping non-scan flops to their scan-equivalent flops.

```
set_db "design:<design_name>" .dft_scan_output_preference auto
```

- This attribute controls which scan flip-flop output pin to use for the scan-data path connection.

```
set_db "design:<design_name>" \
.dft_lockup_element_type preferred_level_sensitive
```

- This attribute controls the type of component to be used as lockup element when combining scan flops triggered by different clocks or different edges of a test clock in the same chain.

```
set_db "design:<design_name>" .dft_mix_clock_edges_in_scan_chains
true
```

- This attribute controls combining flip-flops from the same DFT domain, which are triggered by either edge of the same test clock, on the same scan chain.

```
define_test_clock -name scanclk -period 20000 <clock_port>
```

- This command defines a test clock and associates a test-clock waveform with the clock. The test-clock waveform can be different from the system clocks.

```
define_shift_enable -name se -active high -create_port se
```

- This command specifies the name and active value of the input signal that activates scan shifting. The input signal can be defined on a top-level port or an internal driving pin. This type of input signal is required by the muxed_scan style. The active value of the shift-enable signals is propagated through the design by the check_dft_rules command.

Προσοχή: Ο ορισμός της ιδιότητας (δηλ. του σήματος) shift_enable είναι η μόνη απαραίτητη ιδιότητα που πρέπει να τεθεί ρητώς για την τυπική «εισαγωγή ανίχνευσης» (scan insertion).

```
define_test_mode -name test_mode -active high -create_port test_mode
```

- This command specifies the input signal and constant value that is assigned during a test session. The input signal can be defined on a top-level port or an internal driving pin.

```
define_scan_chain -name top_chain -sdi scan_in -sdo scan_out -
shift_enable se -create_ports
```

- This command creates a scan chain or analyzes an existing chain with the specified input and output scan data ports.

Αφού θέσετε κατάλληλα όλες αυτές τις εντολές, πρέπει να ελέγξετε αν έχουν υπάρξει τυχόν λάθη σε κάποιο από αυτά τα βήματα. Οι παρακάτω εντολές σας βοηθούν να διακριβώσετε, τί ακριβώς έχει συμβεί κατά τη διαδικασία εισαγωγής της λογικής DFT.

```
check_dft_rules
```

```
report_scan_registers
```

```
report_scan_setup
```

- Ελέγξτε αν έχετε παραβάσεις των κανόνων DFT και προσπαθήστε να τις διορθώσετε αν όντως υπάρχουν. Αν δεν υπάρχουν κάποια λάθη, τότε ελέγξτε ξανά αν οι περιορισμοί είναι επαρκείς (το κάνουμε αυτό γιατί έχουμε δημιουργήσει καινούριες θύρες στο κύκλωμά μας!!). Πως μπορείτε να το κάνετε αυτό;
- Μετά από όποιες απαραίτητες διορθώσεις/προσθήκες, το κύκλωμά σας δεν πρέπει να εμφανίζει κανένα πρόβλημα όταν ελέγχετε το χρονισμό του.
- Προχωρήστε στα βήματα της σύνθεσης (syn_generic,...) κλπ.
- Καταγράψτε τα συνήθη αποτελέσματα, gates, area, power, timing (slack). Πως συγκρίνονται τα μεγέθη αυτά για τα βήματα syn_map και syn_opt;
- Ξαναελέγξτε το κύκλωμα σας με την παρακάτω εντολή ελέγχου των κανόνων DFT. Τί πρόβλημα φαίνεται να υπάρχει;

```
check_dft_rules -advanced
```

- Τέλος, πρέπει να συνδέσουμε όλες τις αλυσίδες ανίχνευσης:

```
connect_scan_chains -auto_create_chains
```

```
report_scan_chains
```

Αν θέλετε να παράγετε τις σχετικές αναφορές και αρχεία για το επόμενο στάδιο της ροής σχεδίασης, ακολουθείστε τις παρακάτω εντολές:

```
report_scan_setup
```

```
write_scandef -scanDEF
```

```
write_dft_abstract_model
```

```
write_hdl -abstract
```

```
write_script -analyze_all_scan_chains
```

Παραδοτέα Ασκήσεων 1-9

Για τις παραπάνω ασκήσεις θα πρέπει να υποβάλλετε μία αναφορά η οποία να περιέχει τα ακόλουθα και με την εξής σειρά:

- 1) Το αρχείο περιορισμών .sdc για την Άσκηση 1.
- 2) Τους πίνακες αποτελεσμάτων με κατάλληλες λεζάντες και επεξηγήσεις όπως περιγράφονται στις Ασκήσεις 1 έως 6. Τα ζητούμενα περιγράφονται πιο συγκεκριμένα και σε κάθε μία από αυτές τις ασκήσεις.
- 3) Τα αρχεία .cmd από το genus και το ipnovus για την Άσκηση 1.
- 4) Ό,τι άλλο κρίνεται απαραίτητο το οποίο ανακαλύψατε ή προσθέσατε στην ενασχόλησή σας με τις ασκήσεις αυτές.
- 5) Για την άσκηση 9, απαιτείται να έχετε ολοκληρώσει τη σύνθεση (βήματα 1-6 **μόνο**) χωρίς λάθη και να εξαγάγετε τον αντίστοιχο πίνακα αποτελεσμάτων από το genus. Επίσης, θα πρέπει να μην υπάρχουν παραβάσεις των κανόνων DFT. Δείξτε στην αναφορά σας με τις παραπάνω

εντολές ότι δεν υπάρχουν παραβάσεις των κανόνων DFT και ότι όλες οι αλυσίδες ανίχνευσης έχουν διασυνδεθεί.

Η αναφορά πρέπει να υποβληθεί ως 1 αρχείο pdf όπου θα αναφέρεται το όνομά σας και το ΑΕΜ σας. Σε διαφορετική περίπτωση δε θα δίνεται κανένας βαθμός στην εργασία.

Βαθμολόγηση: Η εργασία είναι ατομική και υποχρεωτική, βαθμολογείται με άριστα το 50 και ισοδυναμεί με το 50% του συνολικού βαθμού του μαθήματος (δηλ. μέχρι 5 βαθμούς). Μη υποβολή ή εκπρόθεσμη υποβολή της εργασίας ισοδυναμεί με απώλεια πέντε βαθμών στο συνολικό βαθμό. Επίσης, η βαθμολόγηση γίνεται με βάση τη συνολική εικόνα της αναφοράς.

Η προθεσμία υποβολής της εργασίας είναι: 10 Φεβρουαρίου 2023 στις 23:59, μόνο μέσω της πλατφόρμας elearning.