

Ψηφιακά Ολοκληρωμένα Κυκλώματα VLSI-ASIC Μεγάλης Κλίμακας

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ

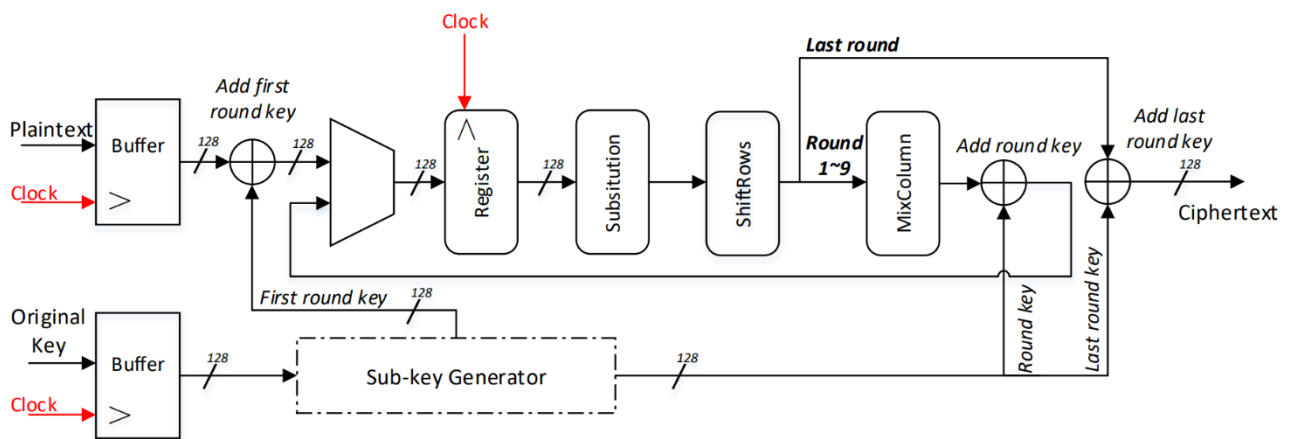
ΑΝ. ΚΑΘΗΓΗΤΗΣ ΒΑΣΙΛΗΣ ΠΑΥΛΙΔΗΣ
ΑΠΟ | ΕΡΓΑΣΤΗΡΙΟ ΗΛΕΚΤΡΟΝΙΚΗΣ

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 κάποιο 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 υλοποιεί (μερικώς) τον αλγόριθμο κρυπτογράφησης που χρησιμοποιείται κατά κόρον σε ηλεκτρονικά προϊόντα και ονομάζεται Advanced Encryption System (AES). Μια υλοποίηση σε υλικό του αλγορίθμου αυτού φαίνεται στο Σχήμα 1. Το κύκλωμα αυτό περιγράφεται στο αρχείο (AES_original.v), όπου κάποια ονόματα αλλάζουν ελαφρώς. Για παράδειγμα, Din = Plaintext και Dout = Ciphertext. Επίσης, σας δίνετε και ένα testbench (AES_TB.v) για να διερευνήσετε τη λειτουργία του κυκλώματος αν το επιθυμείτε.



Σχήμα 1. Υλοποίηση σε υλικό του αλγορίθμου AES.

Άσκηση 1

Σκοπός της άσκησης αυτής είναι να κατανοήσετε τη διαδικασία της σύνθεσης 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 δίνετε λοιπόν ένα απλό κύκλωμα (AES_original.v) και καλείστε εφόσον κατανοήσετε τη λειτουργία του (για το σκοπό αυτό σας δίνετε και το testbench με το οποίο

μπορείτε να προσομοιώσετε το κύκλωμα) να προχωρήσετε στη σύνθεση του κυκλώματος με τη διαδικασία που περιγράφεται στο Εγχειρίδιο του Εργαστηρίου.

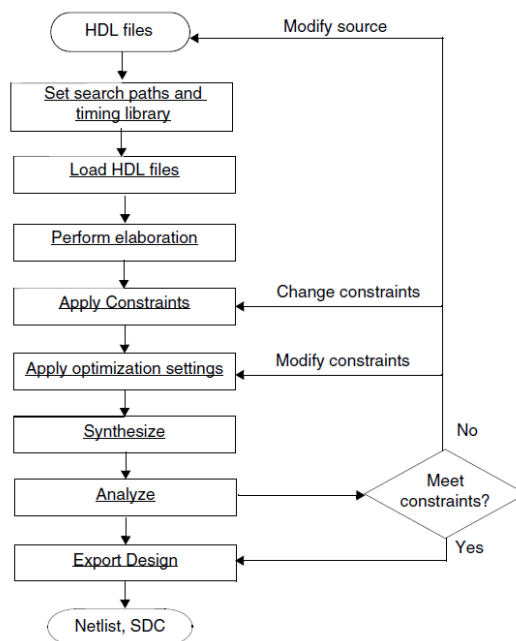
Βήμα 1: Για τη διαδικασία της σύνθεσης, θα πρέπει να ορίσετε τα μονοπάτια για τις βιβλιοθήκες χρονισμού (*.lib). Μερικές βιβλιοθήκες καθυστέρησης ή χρονισμού (timing libraries) βρίσκονται στο μονοπάτι

/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsclib045/timing

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

Στο φάκελο αυτόν επιλέξτε μία βιβλιοθήκη καθυστέρησης που θέλετε να χρησιμοποιήσετε για τη σύνθεση του κυκλώματος. 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: Στην αναφορά σας πρέπει να αναφέρετε ρητώς ποια βιβλιοθήκη επιλέξατε.



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

Αφού θέσετε τα μονοπάτια αυτά, πρέπει να διαβάσετε τη συγκεκριμένη βιβλιοθήκη. Προσέξτε τα μηνύματα που τυχόν παράγει το εργαλείο κατά τη διάρκεια ανάγνωσης της βιβλιοθήκης. Τί παρατηρείτε;

Τέλος, θα πρέπει να θέσετε τα μοντέλα διασυνδέσεων που θα χρησιμοποιηθούν για την ανάλυση χρονισμού στη σύνθεση του κυκλώματος στην επιλογή wireload.

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 μοναδικό φυσικά, 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) τί 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 4 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 300 ps
- 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 ανάλυση χρονισμού setup στο 1 ns όπου θα συμπεριλαμβάνετε 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 ανάλυση χρονισμού hold στο 0.1 ns όπου θα συμπεριλαμβάνετε 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 ανάλυση χρονισμού setup στο 1 ns όπου θα συμπεριλαμβάνετε 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 ανάλυση χρονισμού hold στο 0.1 ns όπου θα συμπεριλαμβάνετε 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 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.

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, ταχύτητα, και ισχύ του κυκλώματος. Τα στοιχεία αυτά θα πρέπει να παρουσιαστούν με τη βοήθεια ενός πίνακα στην αναφορά σας. Επίσης, καλό θα είναι να συμπεριλάβετε και μία περίληψη των αποτελεσμάτων της σύνθεσης και της ποιότητας των αποτελεσμάτων (*quality of results (qor)*). Μπορείτε να παράγετε τα σχετικά αρχεία και να τα αντιγράψετε στην αναφορά σας.

Βήμα 7: Παράγετε όλα τα απαραίτητα αρχεία για τα επόμενα στάδια της ροής σχεδίασης όπου θα χρησιμοποιηθεί το εργαλείο *innovus*

Άσκηση 2

Επαναλάβετε τα βήματα 1-6 για μία συχνότητα ρολογιού 2 ns. Παράγετε τον πίνακα των αποτελεσμάτων της σύνθεσης που περιγράφεται στο Βήμα 6. Δεν είναι απαραίτητη η εξαγωγή περίληψης και ποιότητας αποτελεσμάτων για την άσκηση αυτή.

Άσκηση 3

Επαναλάβετε τα βήματα 1-6 της Άσκησης 1 για μία διαφορετική βιβλιοθήκη από αυτές που είναι διαθέσιμες στο μονοπάτι των βιβλιοθηκών και παράγετε τον πίνακα των αποτελεσμάτων της σύνθεσης που περιγράφεται στο Βήμα 6 της προηγούμενης ενότητας για την καινούρια βιβλιοθήκη. Δεν είναι απαραίτητη η εξαγωγή περίληψης και ποιότητας αποτελεσμάτων για την άσκηση αυτή.

Άσκηση 4

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. Αν και δε θα ασχοληθούμε με το βήμα αυτό στο εργαστήριο αυτό, στην άσκηση αυτή καλείστε να εξοικειωθείτε με το βήμα που μετατρέπει ένα κύκλωμα σε «ικανό προς δοκιμή» κύκλωμα (DFT).

Στην άσκηση αυτή, επομένως, καλείστε για το κύκλωμα που έχουμε δει μέχρι τώρα να εκτελέσετε 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 ελέγεχτε 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 1-4

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 για τις Ασκήσεις 1, 2, και 3 με κατάλληλες λεζάντες και τυχόν επεξηγήσεις αν απαιτούνται
- 3) Την περίληψη αποτελεσμάτων και ποιότητας αποτελεσμάτων για την Άσκηση 1
- 4) Το αρχείο .cmd για την Άσκηση 1
- 5) Ό,τι άλλο κρίνεται απαραίτητο το οποίο ανακαλύψατε ή προσθέσατε στην ενασχόλησή σας με τις ασκήσεις αυτές.
- 6) Για την άσκηση 4 απαιτείται να έχετε ολοκληρώσει τη σύνθεση χωρίς λάθη και να εξάγετε τον αντίστοιχο πίνακα αποτελεσμάτων όπως και στις ασκήσεις 1-3. Επίσης, θα πρέπει να μην υπάρχουν παραβιάσεις των κανόνων DFT.

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 20 και ισοδυναμεί με το 20% του συνολικού βαθμού το μαθήματος (δηλ. μέχρι 2 βαθμούς). Μη υποβολή ή εκπρόθεσμη υποβολή της εργασίας ισοδυναμεί με απώλεια δύο βαθμών στο συνολικό βαθμό. Επίσης, η βαθμολόγηση γίνεται με βάση τη συνολική εικόνα της αναφοράς.

Η προθεσμία υποβολής της εργασίας είναι: 22 Ιανουαρίου 2021 στις 23:59, μόνο μέσω της πλατφόρμας elearning.