

email: grappasv@ece.auth.gr

Αναφορά εργαστηρίου**1) Το αρχείο περιορισμών .sdc για την Άσκηση 1**

Για την **άσκηση 1** δημιουργήθηκε ένα αρχείο **constraints.sdc**, στο οποίο ορίστηκαν οι περιορισμοί που ζητήθηκαν στην άσκηση. Το περιεχόμενο του αρχείου είναι το ακόλουθο:

#1 Ορισμός ρολογιού με 50% duty cycle, περίοδο 4ns και όνομα clk

create_clock -name clk -period 4 -waveform {0 2} [get_ports CLK]

#2 Καθυστέρηση του clk 400ps

set_clock_latency 0.4 [get_clocks {clk}]

#3 Αβεβαιότητα clk 300ps

set_clock_uncertainty 0.3 [get_clocks {clk}]

#4 Χρόνος μετάβασης στο 1% της περιόδου, δηλαδή 0.04ns

set_clock_transition 0.04 -rise [get_clocks clk]

set_clock_transition 0.04 -fall [get_clocks clk]

#5 Καθυστέρηση όλων των εξόδων -> setup 1ns, network latency included.

set_output_delay -max 1 -clock clk [all_outputs] -network_latency_included

#6 Καθυστέρηση όλων των εξόδων -> hold 0.1ns, network latency included.

set_output_delay -min 0.1 -clock clk [all_outputs] -network_latency_included

#7 Φορτίο όλων των εξόδων, setup 0.5pF

set_load 0.5 -max [all_outputs]

#8 Φορτίο όλων των εξόδων, hold 0.01pF

set_load 0.01 -min [all_outputs]

#9 Input delay για όλες τις εισόδους, setup 1ns, network latency included

set_input_delay -max 1 -clock clk [all_inputs] -network_latency_included

#10 Input delay για όλες τις εισόδους, hold 0.1ns, network latency included

set_input_delay -min 0.1 -clock clk [all_inputs] -network_latency_included

#11 Ορισμός κελιών για οδήγηση όλων των εισόδων του κυκλώματος

email: gpappasv@ece.auth.gr

```
set_driving_cell -max -lib_cell BUFX2 [all_inputs]
```

```
set_driving_cell -min -lib_cell BUFX16 [all_inputs]
```

Άσκηση 1

Βήμα 1

Για τις αρχικοποιήσεις του **genus** δημιουργήθηκε ένα αρχείο **run.tcl**, ώστε να γίνονται γρήγορα οι αναθέσεις των paths των Verilog αρχείων, καθώς και του αρχείου sdc. Το περιεχόμενο του run.tcl είναι το ακόλουθο:

```
set_db init_lib_search_path  
/mnt/apps/prebuilt/eda/designkits/GPDK/gsclib045/lan/flow/t1u1/reference_libs/GPDK045/gsclib045_svt_v4.4/gsc  
lib045/timing/
```

```
set_db script_search_path /mnt/scratch_b/users/g/gpappasv/Desktop/Verilog_Files/Scripts/
```

```
set_db init_hdl_search_path /mnt/scratch_b/users/g/gpappasv/Desktop/Verilog_Files/
```

```
set_db library fast_vdd1v0_basicCells.lib
```

```
set_db interconnect_mode wireload
```

Η εντολή `set_db library fast_vdd1v0_basicCells.lib`, μας δείχνει ότι για την πρώτη άσκηση **χρησιμοποιήθηκε η βιβλιοθήκη fast_vdd1v0_basicCells.lib**

(Με μια μικρή έρευνα διαπιστώθηκε ότι γενικά οι fast βιβλιοθήκες προσφέρουν μικρότερες καθυστερήσεις, δηλαδή επιτρέπουν μεγαλύτερους χρονισμούς, με το κόστος βέβαια της υψηλότερης χρήσης ισχύος σε σχέση με τις slow)

Για να ξεκινήσει λοιπόν το genus, χρησιμοποιήθηκε η εντολή **genus -f run.tcl**, ώστε κατά το άνοιγμα του genus να διαβαστεί και το αρχείο run.tcl.

Βήμα 2

Στο βήμα 2 της άσκησης 1, το top-level module θα μπορούσε να προσδιοριστεί με την εντολή **elaborate [top_module_name]**. (Ουσιαστικά έτσι, θα λέγαμε ρητά στο genus, ποιο module θέλουμε να δεχτεί ως top module στην περίπτωση που είχαμε πολλά πιθανά modules.).

Το **top module** του κυκλώματός μας είναι το: **AES_ENC**. Αυτό μας το αναφέρει η εντολή elaborate, όπου στο τέλος της, εμφανίζεται το μήνυμα “Done elaborating ‘AES_ENC’”.

email: grappasv@ece.auth.gr

Αρχικά, με την εντολή **run_hdl AES_original.v**, το genus διάβασε τα αρχεία HDL του κυκλώματος που θέλουμε να συνθέσουμε.

Βήμα 3

Ύστερα, με την εντολή **elaborate**, αναλύεται το κύκλωμα που εισάγαμε στο προηγούμενο βήμα.

Μετά ελέγχθηκε το κύκλωμα με την εντολή **check_design**, όπου δεν εμφανίστηκε κάποιο πρόβλημα.

Name	Total
Unresolved References	0
Empty Modules	0
Unloaded Port(s)	0
Unloaded Sequential Pin(s)	0
Unloaded Combinational Pin(s)	0
Assigns	0
Undriven Port(s)	0
Undriven Leaf Pin(s)	0
Undriven hierarchical pin(s)	0
Multidriven Port(s)	0
Multidriven Leaf Pin(s)	0
Multidriven hierarchical Pin(s)	0
Multidriven unloaded net(s)	0
Constant Port(s)	0
Constant Leaf Pin(s)	0
Constant hierarchical Pin(s)	8328
Preserved leaf instance(s)	0
Preserved hierarchical instance(s)	0
Libcells with no LEF cell	0
Physical (LEF) cells with no libcell	0
Subdesigns with long module name	0
Physical only instance(s)	0
Logical only instance(s)	0

Done Checking the design.

Εικόνα 1: Output της εντολής **check_design**

Βήμα 4

Για την εισαγωγή περιορισμών για τη σύνθεση του κυκλώματος, δημιουργήθηκε το αρχείο **constraints.sdc** που αναφέρθηκε στην αρχή της αναφοράς. Οι περιορισμοί που εφαρμόστηκαν, είναι αυτοί που περιγράφηκαν στο βήμα 4 της άσκησης 1.

Το αρχείο αυτό το διαβάσαμε στο genus με την εντολή **read_sdc constraints.sdc** και δεν εμφανίστηκαν προβλήματα, όπως φαίνεται στην εικόνα που ακολουθεί:

Statistics for commands executed by read_sdc:				
"all_inputs"	- successful	4	, failed	0 (runtime 0.00)
"all_outputs"	- successful	4	, failed	0 (runtime 0.00)
"create_clock"	- successful	1	, failed	0 (runtime 0.00)
"get_clocks"	- successful	4	, failed	0 (runtime 0.00)
"get_ports"	- successful	1	, failed	0 (runtime 0.00)
"set_clock_latency"	- successful	1	, failed	0 (runtime 0.00)
"set_clock_transition"	- successful	2	, failed	0 (runtime 0.00)
"set_clock_uncertainty"	- successful	1	, failed	0 (runtime 0.00)
"set_driving_cell"	- successful	2	, failed	0 (runtime 0.00)
"set_input_delay"	- successful	2	, failed	0 (runtime 0.00)
"set_load"	- successful	2	, failed	0 (runtime 0.00)
"set output delay"	- successful	2	, failed	0 (runtime 0.00)

Εικόνα 2: Output της εντολής **read_sdc constraints.sdc**

Παράλληλα, χρησιμοποιήθηκε η εντολή **check_timing_intent** για να ελεγχθεί αν ο χρονισμός του κυκλώματος και οι περιορισμοί που υλοποιήθηκαν, περιγράφηκαν χωρίς προβλήματα. Το αποτέλεσμα της εντολής αυτής έδειξε ότι δεν υπήρχε κανένα πρόβλημα.

email: gpappasv@ece.auth.gr

```

Lint summary
Unconnected/logic driven clocks                                0
Sequential data pins driven by a clock signal                  0
Sequential clock pins without clock waveform                   0
Sequential clock pins with multiple clock waveforms           0
Generated clocks without clock waveform                       0
Generated clocks with incompatible options                     0
Generated clocks with multi-master clock                      0
Paths constrained with different clocks                        0
Loop-breaking cells for combinational feedback                0
Nets with multiple drivers                                    0
Timing exceptions with no effect                               0
Suspicious multi cycle exceptions                             0
Pins/ports with conflicting case constants                     0
Inputs without clocked external delays                        0
Outputs without clocked external delays                       0
Inputs without external driver/transition                     0
Outputs without external load                                  0
Exceptions with invalid timing start-/endpoints                0
Total:                                                         0

```

Εικόνα 3: Output της εντολής `check_timing_intent`

Βήμα 5

Σε αυτό το βήμα χρησιμοποιήθηκαν οι εντολές **syn_generic**, για τη σύνθεση του κυκλώματος, **syn_map** και **syn_opt**, για να βελτιστοποιηθεί η σύνθεση.

Βήμα 6

Αφού έγινε η σύνθεση, χρησιμοποιήθηκαν οι εντολές **report_area**, **report_gates**, **report_timing**, **report_power**, **report_qor > my_qor.txt**, για να παρουσιαστούν τα χαρακτηριστικά του κυκλώματος που θα παρουσιαστούν αργότερα.

Οι ασκήσεις 2 και 3 έχουν ίδια μεθοδολογία, με μικρές διαφορές, οπότε θα παρουσιαστούν μόνο τα αποτελέσματα/χαρακτηριστικά του κυκλώματος που δημιουργήθηκε, όπως ζητείται.

2) Τους πίνακες αποτελεσμάτων για τις ασκήσεις 1, 2, 3:

Άσκηση 1

Instance	Module	Cell	Count	Cell Area	Net Area	Total Area	Wireload
AES_ENC			7706	14456.682	0	14456.682	<none>

Πίνακας 1: Αναφορά επιφάνειας (*report area*)

Όπως βλέπουμε στον πίνακα 1, χρησιμοποιήθηκαν 7706 κελιά με συνολική επιφάνεια 14456,682.

Type	Instances	Area	Area %
sequential	386	2374.85	16.4
inverter	792	542.412	3.8
buffer	130	1067.04	7.4
logic	6398	10472.4	72.4
physical_cells	0	0	0
total	7706	14456.7	100

Πίνακας 2: Πύλες που χρησιμοποιήθηκαν (*report_gates*) (μαζί με την επιφάνεια που χρησιμοποιεί η κάθε πύλη)

email: grappasv@ece.auth.gr

Στον πίνακα 2 βλέπουμε πόσες πύλες από τον κάθε τύπο χρησιμοποιήθηκαν, καθώς και την επιφάνεια που καταλαμβάνει ο κάθε τύπος πυλών (πχ έχουμε 130 buffers που κατέλαβαν επιφάνεια 1067,04)

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
register	1.36353E-07	3.98185E-04	4.19267E-05	4.40248E-04	9.13%
latch	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
logic	8.50474E-07	5.16334E-04	3.83082E-03	4.34800E-03	90.15%
bbox	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
clock	0.00000E+00	0.00000E+00	3.50295E-05	3.50295E-05	0.73%
pad	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
pm	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
Subtotal	9.86827E-07	9.14519E-04	3.90778E-03	4.82328E-03	100.01%
Percentage	0.02%	18.96%	81.02	100.00%	100.00%

Πίνακας 3: Αναφορά ισχύος του κυκλώματος (report_power)

Timing Point	Flags	Arc	Edge	Cell	Fanout	Load (fF)	Trans (ps)	Delay (ps)	Arrival (ps)	Instance Location
BSYrg_reg/CK	-	-	R	(arrival)	386	-	40	-	400	(-, -)
BSYrg_reg/Q	-	CK->Q	F	DFFHQX1	3	2.9	22	65	465	(-, -)
fopt/Y	-	A->Y	F	BUFX20	1	500.0	202	138	603	(-, -)
BSY	<<<	-	F	(port)	-	-	-	0	603	(-, -)
Output Delay:-	1000	Uncertainty:-	300	Required Time:=	2700	Launch Clock:-	400	Data Path:-	203	Slack:=2097

Πίνακας 4: Αναφορά ταχύτητας του κυκλώματος (report_timing)

Στον πίνακα 4 βλέπουμε αναλυτικά το πιο αργό μονοπάτι του κυκλώματος. Και απο το κάτω μέρος του πίνακα εξάγουμε την τιμή του slack ως **slack = Tclk – Output_Delay – Uncertainty – Required_Time – Launch_Clock – Data_Path = 2097**

Area	Gate#	Total Power	Slack
14456.682	7706	4.82328E-03	2097

Πίνακας 5: Συγκεντρωτικά αποτελέσματα άσκησης 1

Στον πίνακα 5 βλέπουμε συνοπτικά μία χρήσιμη σύνοψη των παραπάνω αποτελεσμάτων.

Άσκηση 2

Instance	Module	Cell	Count	Cell Area	Net Area	Total Area	Wireload
AES_ENC			8059	14738.490	0	14738.490	<none>

Πίνακας 6: Αναφορά επιφάνειας (report_area)

Type	Instances	Area	Area %
sequential	386	2374.848	16.1
inverter	919	628.938	4.3
buffer	130	1067.04	7.2
logic	6623	10664.586	72.4

email: gpappasv@ece.auth.gr

physical_cells	0	0	0
total	8058	14735.412	100

Πίνακας 7: Πύλες που χρησιμοποιήθηκαν (report_gates) (μαζί με την επιφάνεια που χρησιμοποιεί η κάθε πύλη)

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
register	1.38154E-07	7.54145E-04	7.22198E-05	8.26503E-04	9.23%
latch	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
logic	8.71153E-07	8.50261E-04	7.20792E-03	8.05905E-03	89.99%
bbox	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
clock	0.00000E+00	0.00000E+00	7.00590E-05	7.00590E-05	0.78%
pad	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
pm	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
Subtotal	1.00931E-06	1.60441E-03	7.35020E-03	8.95561E-03	100%
Percentage	0.01%	17.92%	82.07%	100.00%	100.00%

Εδώ βλέπουμε ξεκάθαρα (στην τιμή Total, του Subtotal) ότι η κατανάλωση αυξήθηκε αρκετά σε σχέση με την άσκηση 1, αφού πλέον δουλεύουμε το κύκλωμα με 2 φορές πιο γρήγορο ρολόι, κάτι που ήταν αναμενόμενο.

Πίνακας 8: Αναφορά ισχύος του κυκλώματος (report_power)

Timing Point	Flags	Arc	Edge	Cell	Fanout	Load (fF)	Trans (ps)	Delay (ps)	Arrival (ps)	Instance Location
BSYrg_reg/CK	-	-	R	(arrival)	386	-	20	-	400	(-, -)
BSYrg_reg/Q	-	CK->Q	F	DFFHQX1	3	2.7	20	61	461	(-, -)
g32082/Y	-	A->Y	F	BUFX20	1	500.0	202	138	599	(-, -)
BSY	<<<	-	F	(port)	-	-	-	0	599	(-, -)
Output				Required		Launch		Data		
Delay:-	1000	Uncertainty:-	300	Time:=	700	Clock:-	400	Path:-	199	Slack:=101

Πίνακας 9: Αναφορά ταχύτητας του κυκλώματος (report_timing)

Όπως βλέπουμε στο slack στον πίνακα 9 σε αυτή την περίπτωση, μειώνοντας το ρολόι στη μισή περίοδο (από 4000 σε 2000 ps), το slack μειώθηκε επίσης λίγο - πολύ κατά 2000 κάτι που ήταν αναμενόμενο γιατί το slack μας φανερώνει το πόσα περιθώρια έχουμε για μείωση της περιόδου ρολογιού. Εμείς τώρα πρακτικά εκμεταλλευτήκαμε πλήρως τα περιθώρια ταχύτητας του ολοκληρωμένου (το να πλησιάζουμε όμως στο Slack = 0 όμως δεν είναι θεμιτό καθώς οποιαδήποτε διακύμανση στις καθυστερήσεις θα οδηγούσε σε αστάθεια) Έχοντας ακόμα θετικό slack, μπορούμε να το τρέξουμε το ολοκληρωμένο στην ταχύτητα που του ζητήσαμε.

Area	Gate#	Total Power	Slack
14735.412	8058	8.95561E-03	101

Πίνακας 10: Συγκεντρωτικά αποτελέσματα άσκησης 2

Άσκηση 3

Η βιβλιοθήκη που χρησιμοποιήθηκε σε αυτή την άσκηση ήταν η **slow_vdd1v2_basicCells.lib**

Instance	Module	Cell	Count	Cell Area	Net Area	Total Area	Wireload
----------	--------	------	-------	-----------	----------	------------	----------

email: gpappasv@ece.auth.gr

AES_ENC	8142	14802.786	0	14802.786	<none>
---------	------	-----------	---	-----------	--------

Πίνακας 11: Αναφορά επιφάνειας (report_area)

Type	Instances	Area	Area %
sequential	386	2374.848	16
inverter	906	620.388	4.2
buffer	130	1067.04	7.2
logic	6720	10740.51	72.6
physical_cells	0	0	0
total	8142	14802.786	100

Πίνακας 12: Πύλες που χρησιμοποιήθηκαν (report_gates) (μαζί με την επιφάνεια που χρησιμοποιεί η κάθε πύλη)

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
register	7.56002E-08	3.47298E-04	3.10684E-05	3.78442E-04	8.82%
latch	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
logic	5.59469E-07	4.27932E-04	3.46125E-03	3.88974E-03	90.66%
bbox	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
clock	0.00000E+00	0.00000E+00	2.25115E-05	2.25115E-05	0.52%
pad	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
pm	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
Subtotal	6.35069E-07	7.75230E-04	3.51483E-03	4.29069E-03	100.00%
Percentage	0.01%	18.07%	81.92%	100.00%	100.00%

Πίνακας 13: Αναφορά ισχύος του κυκλώματος (report_power)

Timing Point	Flags	Arc	Edge	Cell	Fanout	Load (fF)	Trans (ps)	Delay (ps)	Arrival (ps)	Instance Location
BSYrg_reg/CK	-	-	R	(arrival)	386	-	40	-	400	(-, -)
BSYrg_reg/Q	-	CK->Q	F	DFFHQX1	3	2.6	45	143	543	(-, -)
drc_bufs116109/Y	-	A->Y	F	CLKBUF20	1	500.0	412	284	827	(-, -)
BSY	<<<	-	F	(port)	-	-	-	0	827	(-, -)
Output Delay:-	1000	Uncertainty:-	300	Required Time:=	2700	Launch Clock:-	400	Data Path:-	427	Slack:=1873

Πίνακας 14: Αναφορά ταχύτητας του κυκλώματος (report_timing)

Area	Gate#	Total Power	Slack
14802.786	8142	4.29069E-03	1873

Πίνακας 15: Συγκεντρωτικά αποτελέσματα άσκησης 3

Όπως βλέπουμε στον πίνακα 15, το slack είναι μειωμένο σε σχέση με αυτό που είχαμε στην άσκηση 1. Αυτό μας λέει πως το ολοκληρωμένο μας, έχει μικρότερα περιθώρια αύξησης της ταχύτητας του ρολογιού χρησιμοποιώντας τη βιβλιοθήκη **slow_vdd1v2_basicCells.lib**, σε σχέση με τη βιβλιοθήκη **fast_vdd1v0_basicCells.lib**. Ωστόσο βλέπουμε ότι το total power παρουσίασε, στην περίπτωση της slow βιβλιοθήκης, μια μείωση συγκριτικά με τη fast. Οπότε, ανάλογα τις απαιτήσεις που έχουμε από το ολοκληρωμένο, θα προτιμήσουμε τη μία ή την άλλη βιβλιοθήκη.

3) Περίληψη αποτελεσμάτων και ποιότητας αποτελεσμάτων για την άσκηση 1

email: grappasv@ece.auth.gr

Για την εξαγωγή του αρχείου που χρειάζεται γι αυτό το ερώτημα, χρησιμοποιήθηκε η εντολή **report qor** > **my_qor.txt**. Το αποτέλεσμα του αρχείου αυτού είναι το ακόλουθο:

```
Timing
-----

Clock Period
-----
clk    4000.0

Cost      Critical    Violating
Group     Path Slack   TNS      Paths
-----
clk       2096.6    0.0      0
default   No paths    0.0
-----
Total                    0.0      0

Instance Count
-----
Leaf Instance Count          7706
Physical Instance count      0
Sequential Instance Count    386
Combinational Instance Count 7320
Hierarchical Instance Count  0

Area
----
Cell Area                    14456.682
Physical Cell Area           0.000
Total Cell Area (Cell+Physical) 14456.682
Net Area                     0.000
Total Area (Cell+Physical+Net) 14456.682

Max Fanout                    386 (CLK)
Min Fanout                     1 (BSY)
Average Fanout                 2.9
Terms to net ratio             3.8142
Terms to instance ratio       3.9429
Runtime                       340.999995 seconds
Elapsed Runtime               4530 seconds
Genus peak memory usage       983.93
Innovus peak memory usage     no_value
Hostname                      cn2.it.auth.gr
```

Εικόνα 4: Quality of Results

Για την εξαγωγή της περίληψης, χρησιμοποιήθηκε η εντολή **report summary** > **sum.txt**

Το περιεχόμενο του αρχείου είναι το ακόλουθο:

```
=====
Generated by:    Genus(TM) Synthesis Solution 19.11-s087_1
Generated on:    Jan 06 2021 01:07:04 am
Module:          AES_ENC
Operating conditions: PVT_1P1V_OC (balanced_tree)
Wireload mode:   enclosed
Area mode:       timing library
=====
```

Timing

email: gpappasv@ece.auth.gr
Slack Endpoint Cost Group

+2097ps BSY clk

Area

Instance	Module	Cell Count	Cell Area	Net Area	Total Area	Wireload
----------	--------	------------	-----------	----------	------------	----------

AES_ENC		7706	14456.682	0.000	14456.682	<none> (D)
---------	--	------	-----------	-------	-----------	------------

(D) = wireload is default in technology library

Design Rule Check

Max_transition design rule: no violations.

Max_capacitance design rule (violation total = 26000.0)

Worst violator:

Pin	Load (ff)	Max	Violation
fopt238863/Y	500.0	300.0	200.0

4) Το αρχείο cmd για την άσκηση 1:

genus.cmd

Cadence Genus(TM) Synthesis Solution, Version 19.11-s087_1, built Aug 13 2019 11:48:07

Date: Wed Jan 06 00:56:08 2021

Host: cn2.it.auth.gr (x86_64 w/Linux 3.10.0-1160.6.1.el7.x86_64) (10cores*20cpus*2physical
cpus*Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz 25600KB)

OS: CentOS Linux release 7.9.2009 (Core)

source run.tcl

read_hdl AES_original.v

elaborate

check_design

read_sdc constraints.sdc

check_timing_intent

syn_generic

syn_map

```
email: gpappasv@ece.auth.gr
syn_opt
report_area
report_gates
report_timing
report_power
write_hdl > finalAES.v
write_script > constr.g
write_sdc > constr.sdc
write_design -innovus AES_ENC
report qor > my_qor.txt
report summary > sum.txt
write_template -h
write_template -split -performance -outfile template.g
quit
```

6) Πίνακας αποτελεσμάτων για την άσκηση 4 και παρουσίαση της άσκησης:

Άσκηση 4 (Χρησιμοποιήθηκε η βιβλιοθήκη **fast_vdd1v0_basicCells.lib**, όπως και στην πρώτη άσκηση)

Σκοπός της άσκησης 4 είναι να εξοικειωθούμε με το βήμα της μετατροπής ενός κυκλώματος σε «ικανό προς δοκιμή» κύκλωμα (**DFT**). Αυτή η μετατροπή έχει να κάνει με την προσθήκη λογικής στο κύκλωμα, που το καθιστά κατάλληλο για να ελεγχθεί από εξοπλισμό ελέγχου μετά την κατασκευή του ολοκληρωμένου, ενώ ταυτόχρονα, αυτή η επιπλέον λογική, δε μεταβάλλει τη λειτουργία του ολοκληρωμένου. Για να εισαχθεί η λογική αυτή στο ολοκληρωμένο μέσω του **genus**, χρησιμοποιήθηκαν οι εντολές που ζητήθηκαν στο pdf της άσκησης, κάτι που φαίνεται από το περιεχόμενο του αρχείου **genus.cmd**:

```
# Cadence Genus(TM) Synthesis Solution, Version 19.11-s087_1, built Aug 13 2019 11:48:07
```

```
# Date: Wed Jan 06 16:02:50 2021
```

```
# Host: cn2.it.auth.gr (x86_64 w/Linux 3.10.0-1160.6.1.el7.x86_64) (10cores*20cpus*2physical cpus*Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz 25600KB)
```

```
# OS: CentOS Linux release 7.9.2009 (Core)
```

```
source run.tcl
read_hdl AES_original.v
elaborate
set_db /.dft_scan_style muxed_scan
set_db /.dft_prefix DFT_
set_db /.dft_identify_top_level_test_clocks true
set_db /.dft_identify_test_signals true
set_db /.dft_identify_internal_test_clocks false
set_db /.use_scan_seqs_for_non_dft false
set_db "design:AES_ENC".dft_scan_map_mode tdr_pass
set_db "design:AES_ENC"\
.dft_connect_shift_enable_during_mapping tie_off
set_db "design:AES_ENC"\
.dft_connect_scan_data_pins_during_mapping loopback
set_db "design:AES_ENC".dft_scan_output_preference auto
```

email: gpappasv@ece.auth.gr

```
set_db "design:AES_ENC" \  
.dft_lockup_element_type preferred_level_sensitive  
set_db "design:AES_ENC" .dft_mix_clock_edges_in_scan_chains true  
define_test_clock -name scanclk -period 20000 CLK  
define_shift_enable -name se -active high -create_port se  
define_test_mode -name test_mode -active high -create_port test_mode  
define_scan_chain -name top_chain -sdi scan_in -sdo scan_out -shift_enable se -create_ports  
check_dft_rules  
report_scan_registers  
report_scan_setup  
check_timing_intent  
read_sdc constraints.sdc  
check_timing_intent  
syn_generic  
syn_map  
report_area  
report_gates  
report_timing  
report_timing  
syn_opt  
report_area  
report_gates  
report_timing  
report_power  
check_dft_rules -advanced  
check_design -undriven -report_scan_pins  
connect_scan_chains -auto_create_chains  
check_dft_rules -advanced  
check_design -undriven -report_scan_pins  
report_scan_chains  
report_scan_setup  
write_scandef -scanDEF  
write_dft_abstract_model  
write_hdl -abstract  
write_script -analyze_all_scan_chains  
quit
```

Μετά την προσθήκη των εντολών για την εισαγωγή λογικής που προσφέρει DFT στο ολοκληρωμένο, χρησιμοποιήθηκαν, όπως φαίνεται στο `genus.cmd` που παρουσιάστηκε προηγουμένως, οι εντολές **check_dft_rules**, **report_scan_registers**, **report_scan_setup**, για να ελεγχθεί αν υπάρχουν τυχόν λάθη/προβλήματα στο κύκλωμα, μετά από αυτές.

Τα **αποτελέσματα** που επέστρεψαν αυτές οι εντολές φαίνονται στη συνέχεια:

email: gpappasv@ece.auth.gr

```
@genus:root: 20> check_dft_rules
Checking DFT rules for 'AES_ENC' module under 'muxed_scan' style

Checking DFT rules for clock pins
... Processed 250 registers
Checking DFT rules for async. pins
... Processed 250 registers
Checking DFT rules for shift registers.
Detected 0 DFT rule violation(s)
Summary of check_dft_rules
*****
Number of usable scan cells: 48
Clock Rule Violations:
-----
Internally driven clock net: 0
Tied constant clock net: 0
Undriven clock net: 0
Conflicting async & clock net: 0
Misc. clock net: 0

Async. set/reset Rule Violations:
-----
Internally driven async net: 0
Tied active async net: 0
Undriven async net: 0
Misc. async net: 0

Total number of DFT violations: 0

Total number of Test Clock Domains: 1
Number of user specified non-Scan registers: 0
Number of registers that fail DFT rules: 0
Number of registers that pass DFT rules: 386
Percentage of total registers that are scannable: 100%
0
```

Εικόνα 5: Output της εντολής `check_dft_rules`

```
Summary:
Total registers that pass DFT rules: 386
Total registers that fail DFT rules: 0
Total registers that are marked preserved or dont-scan: 0
Total registers that are marked Abstract Segment dont-scan: 0
Total registers that are part of shift register segments: 0
Total registers that are lockup elements: 0
Total registers that are level-sensitive: 0
Total registers that are misc. non-scan: 0
```

Εικόνα 6: Output της εντολής `report_scan_registers`

```
@genus:root: 22> report_scan_setup

Design Name
=====
    AES_ENC

Scan Style
=====
    muxed_scan

Design has a valid DFT rule check status
```

Εικόνα 7: Output της εντολής `report_scan_setup`

Από τις παραπάνω εικόνες, μπορούμε να διαπιστώσουμε ότι δεν προέκυψε κάποιο πρόβλημα στο κύκλωμα με την προσθήκη της επιπλέον λογικής. Στη συνέχεια προστέθηκαν και οι περιορισμοί στο κύκλωμα, με την εντολή `read_sdc constraints.sdc`, όπως και στην άσκηση 1. Στο συγκεκριμένο βήμα, αν διαβάζαμε τα constraints πριν εισαχθεί η επιπλέον λογική στο κύκλωμα, θα διαπιστώναμε ότι με την εντολή `check_timing_intent` θα εμφανιζόντουσαν προβλήματα. Αυτό οφείλεται στο ότι οι περιορισμοί δε θα είχαν εφαρμοστεί στην επιπλέον λογική, κάτι που λύνεται απλά διαβάζοντας τους κανόνες αφότου εισαχθεί το επιπλέον κύκλωμα.

email: grappasv@ece.auth.gr

Έτσι, χρησιμοποιώντας τώρα την εντολή **check_timing_intent** για τον έλεγχο του χρονισμού του κυκλώματος, δεν παρουσιάστηκε κάποιο πρόβλημα.

Στη συνέχεια ακολούθησαν οι εντολές **syn_generic**, **syn_map**, **syn_opt** για τη σύνθεση του κυκλώματος.

Μετά, όπως και στις προηγούμενες ασκήσεις, καταγράφηκαν τα ζητούμενα **αποτελέσματα** και παρουσιάζονται στη συνέχεια:

Instance	Module	Cell	Count	Cell Area	Net Area	Total Area	Wireload
AES_ENC			7814	15261.408	0	15261.408	<none>

Πίνακας 16: Αναφορά επιφάνειας (report area)

Type	Instances	Area	Area %
sequential	386	2904.264	19
inverter	792	542.07	3.6
buffer	130	1067.04	7
logic	6506	10748.034	70.4
physical_cells	0	0	0
total	7814	15261.408	100

Πίνακας 17: Πύλες που χρησιμοποιήθηκαν (report_gates) (μαζί με την επιφάνεια που χρησιμοποιεί η κάθε πύλη)

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
register	1.50996E-07	3.94435E-04	4.51848E-05	4.39771E-04	9.15%
latch	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
logic	8.69782E-07	5.19558E-04	3.81231E-03	4.33274E-03	90.12%
bbox	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
clock	0.00000E+00	0.00000E+00	3.50295E-05	3.50295E-05	0.73%
pad	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
pm	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00%
Subtotal	1.02078E-06	9.13993E-04	3.89252E-03	4.80754E-03	100.00%
Percentage	0.02%	19.01%	80.97%	100.00%	100.00%

Πίνακας 18: Αναφορά ισχύος του κυκλώματος (report_power)

Timing Point	Flags	Arc	Edge	Cell	Fanout	Load (fF)	Trans (ps)	Delay (ps)	Arrival (ps)	Instance Location
BSYrg_reg/CK	-	-	R	(arrival)	386	-	40	-	400	(-, -)
BSYrg_reg/Q	-	CK->Q	F	SDFQX1	4	3.1	23	63	463	(-, -)
fopt/Y	-	A->Y	F	BUFX20	1	500.0	202	139	602	(-, -)
BSY	<<<	-	F	(port)	-	-	-	0	602	(-, -)
Output Delay:-	1000	Uncertainty:-	300	Required Time:=	2700	Launch Clock:-	400	Data Path:-	202	Slack:= 2098

Πίνακας 19: Αναφορά ταχύτητας του κυκλώματος (report_timing)

Μετά, με τη χρήση της εντολής **check_dft_rules -advanced**, παρουσιάζεται το **warning** ότι υπάρχει ένα μη οδηγούμενο pin, που μπορεί να λειτουργήσει σαν x-source generator.

email: grappasv@ece.auth.gr

Ενώ βλέπουμε πάλι πως δεν υπάρχει κανένα DTF violation:

Total number of DFT violations: 0

Μετα τη χρήση της εντολής **connect_scan_chains -auto_create_chains**, για τη δημιουργία των αλυσίδων ανίχνευσης, αν τρέξουμε πάλι την εντολή **check_dft_rules -advanced**, θα διαπιστώσουμε ότι το παραπάνω **warning**, δεν εμφανίζεται πλέον.

Σύγκριση των αποτελεσμάτων των **gates/area/power/timing (slack)** για τα βήματα **syn_map** και **syn_opt**:

Area	Gate#	Total Power	Slack
15261.408	7814	4.80754E-03	2098

Πίνακας 20: Συγκεντρωτικά αποτελέσματα μετά τις εντολές **syn_generic, syn_map, syn_opt**

Area	Gate#	Total Power	Slack
15293.556	7841	4.80885E-03	2098

Πίνακας 21: Συγκεντρωτικά αποτελέσματα μετά τις εντολές **syn_generic, syn_map** και πριν την **syn_opt**

Όπως βλέπουμε απο τον πίνακα 20, η εντολή **syn_opt**, μειώνει ελαφρώς τον αριθμό των πυλών, και ως εκ τούτου βλέπουμε και μια ανάλογη (ελάχιστη) μείωση στην κατανάλωση και στο Area, κάτι που είναι λογικό, αφού η εντολή **syn_opt** έχει ως σκοπό την εφαρμογή αλγορίθμων για να επιτευχθεί μια επιπλέον βελτιστοποίηση, όπου είναι δυνατό.