



#### Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών Πανεπιστήμιο Θεσσαλίας

## ΕCE431 - Αλγόριθμοι CAD Ι - Λογική Σύνθεση

Χειμερινό Εξάμηνο — Ακαδημαϊκό Έτος 2023-2024

#### Εργαστηριακή Εργασία 4<sup>η</sup>

## Working with BDDs using CUDD package

22/1/2024 έως 25/2/2024

Χ. Σωτηρίου

## 1 Στόχοι 4ης Εργασίας

Οι στόχοι της 4ης εργασίας του μαθήματος είναι:

- 1. Η κατασκευή BDD για κάθε πύλη ενός design συναρτήσει των Primary Inputs του.
- 2. Η ανάθεση πιθανοτήτων σε όλα τα gatepins του design (Static Probabilities Annotation).

# 2 Μερος A: BDD Construction for each Circuit Instance

Σε αυτό το μέρος της εργασίας θα πρέπει να κατασκευάσετε ένα BDD για κάθε μία πύλη του κυκλώματος. Προσοχή το BDD που προκύπτει θα πρέπει να είναι συνάρτηση των BDDs των εξόδων των πυλών που οδηγούν τις εισόδους της εκάστοτε πύλης. Αυτό σημαίνει ότι για κάθε πύλη που οι είσοδοι της συνδέονται με την έξοδο άλλων πυλών του κυκλώματος πρέπει πρώτα να υπολογιστεί το BDD για εκείνες τις πύλες και κατόπιν να υπολογιστεί την επικείμενη πύλη. Γιαυτό τον λόγο είναι απαραίτητο να αξιοποιήσετε το Levelization των gatepins που έχετε υλοποιήσει στην προηγούμενη εργασία.

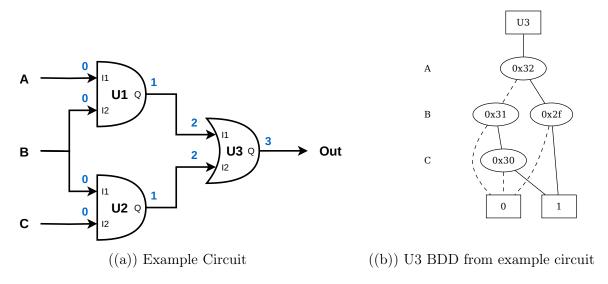


Figure 1: Circuit and BDD Demostration

Όπως φαίνεται και από το παράδειγμα χωρίς τα BDD των predecessors δεν μπορύμε να κατασκευάσουμε πιο μπροστά τα διαγραμμάτα. Γιαυτό το λόγο σας προτείνεται να αξιοποιήσετε το Levelisation που έχετε υλοποιήσει στην προηγούμενη εργασία, ώστε να διατρέχετε το design level by level και να γνωρίζετε ότι για κάθε πύλη που επισκέπτεστε έχουν υπολογιστεί όλα τα απαραίτητα προηγούμενα διαγράμματα. Τα gatepins στο επίπεδο μηδέν (Startpoints) είναι πάντα Primary Inputs, Constants ή Flip-Flop outputs και ορίζουν τις μεταβλητές των BDDs που θα κατασκευάσετε για τα επόμενα επίπεδα. Σας προτείνεται να κρατάτε μια λίστα με τη σειράς την οποία προσθέτετε μια μεταβλητή στον Manager, καθώς αυτό θα σας χρειαστεί στο επόμενο μέρος της εργασίας, για να μπορέσετε να εντοπίσετε ποιοι κόμβοι ενός BDD αντιστοιχούν σε κάθε μεταβλητή.

Σημείωση: BDD σας ζητείται να κατασκευάσετε μόνο για τα output gatepins, τα διαγράμματα των input gapins καθορίζονται από το driver output gatepin.

Ενδειχτικά οι TCL εντολές που καλείστε να υλοποιήσετε ή να επεχτείνετε είναι οι εξής:

- report\_component\_BDD <component name>
- report\_gatepin\_BDD <gatepin name>

Οι οποίες θα εμφανίζουν στον χρήστη το BDD του component ή του gatepin βάση του spec που περιγράφηκε πιο πάνω. Για παράδειγμα, με την εντολή report\_component\_BDD U3 στο εργαλείο σας, για το κύκλωμα της εικόνας 1(a), θα πάρω το διάγραμμα της εικόνας 1(b).

## 3 Μέρος B: Static Probabilities Annotation

Σε αυτό το μέρος της εργασίας καλείστε να υπολογίσετε την πιθανότητα κάθε gatepin του κυκλώματος να είναι στην κατάσταση logic-1 ή logic-0. Για να το κάνετε αυτό θα βασιστείτε στα BDDs που θα έχετε υπολογίσει στο προηγούμενο μέρος της εργασίας. Θα έχετε λοιπόν για κάθε πύλη του κυκλώματος τη συνάρτηση εξόδου συναρτήσει των Startpoints του design και έτσι ο υπολογισμός της πιθανότητας η πύλη να βρίσκεται πχ. στην κατάσταση logic-1 γίνεται πολύ εύκολα βρίσκοντας όλα τα μονοπάτια από τη ρίζα προς το φύλο του logic-1. Περισσότερες πληροφορίες και παράδειγμα θα βρείτε στη σχετική διάλεξη του μαθήματος.

Ο χρήστης του εργαλείου σας θα σας ορίσει τις πιθανότητες για τα Startpoints μέσω της ακόλουθης TCL εντολής που καλείστε να υλοποιήσετε:

• set\_static\_probability -value <probability value> -gatepins {gatepins list} | -allstartpoints

Στην οποία το όρισμα -value παίρνει μετά έναν double που ορίζει την πιθανότητα ένα Startpoint να είναι στην κατάσταση logic-1. Το -gatepins παίρνει μετά μια λίστα από gatepins στα οποία θέλει ο χρήστης να αναθέσει την πιθανότητα. Αν ο χρήστης ορίσει -allstartpoints τότε η παράμετρος -gatepins δε χρειάζεται να οριστεί, και το εργαλείο σας θα αναθέσει την πιθανότητα που διάβασε σε όλα τα Startpoints του design. Αν δε δοθούν οι πιθανότητες τότε το εργαλείο σας θα θεωρεί 50% για όλα τα level 0 gatepins.

Υπενθύμιση, δε χρειάζεται να υπολογίσετε ξεχωριστά τις πιθανότητες για logic-1 και logic-0, βρίσκοντας την πιθανότητα για τη μία κατάσταση (πχ για logic-1 ( $P_1$ )) τότε η άλλη είναι η διαφορά από το 1 (δηλαδή  $P_0=1-P_1$ ).

Η σχετική ΤCL εντολή που καλείστε να υλοποιήσετε για αυτό το μέρος της εργασίας είναι η εξής:

• list\_static\_probabilities -gatepins {gatepins list} | -allgatepins

Η οποία θα παίρνει είτε μία λίστα από gatepins μετά την παράμετρο -gatepins είτε την παράμετρο -allgatepins και θα τυπώνει τις πιθανότητες για logic-1 και logic-0 για τη λίστα είτε όλα τα gatepins.

## 3.1 CUDD Tips

Κάνοντας include τη βιβλιοθήκη "cuddInt.h" σαν επιτρέπει να χρησιμοποιήσετε το πακέτο CUDD ως clear box και να έχετε πρόσβαση στις εσωτερικές δομές του. Με αυτόν τον τρόπο θα μπορείτε να κάνετε traverse το BDD και να βρείτε όλα τα μονοπάτια προς το λογικό 1 ή 0, ώστε να υπολογίσετε τις αντίστοιχες πιθανότητες. Οι βασικές δομές που σας ενδιαφέρουν είναι οι εξής:

- DdManager: Central data structure of CUDD.
- DdNode: Decision diagram node.
- DdChildren: The two children of a non-terminal node.

Καθώς επίσης και όποιες άλλες εσείς θεωρείτε ότι σας είναι απαραίτητες για την υλοποίηση σας. Τέλος, συστήνεται να δείτε τη συνάρτηση Cudd\_DumpDot() ώστε να καταλάβετε πως αντιστοιχίζονται οι κόμβοι ενός διαγράμματος με τα variables για να ξέρετε την πιθανότητα του κάθε edge κατά το traversal του BDD.

## 4 Παρουσίαση και Demo

Προετοιμάστε ένα σύντομο ppt και μια επίδειξη του εκτελέσιμου για την ώρα του μαθήματος. Η προθεσμία υποβολής 4ης Εργασίας είναι η 25/2/2024. Μέχρι τότε θα πρέπει να έχετε υποβάλλει τα αρχεία της εργασίας στο e-Class