



Πανεπιστήμιο Αιγαίου

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Εισαγωγή σε VLSI

3η Άσκηση

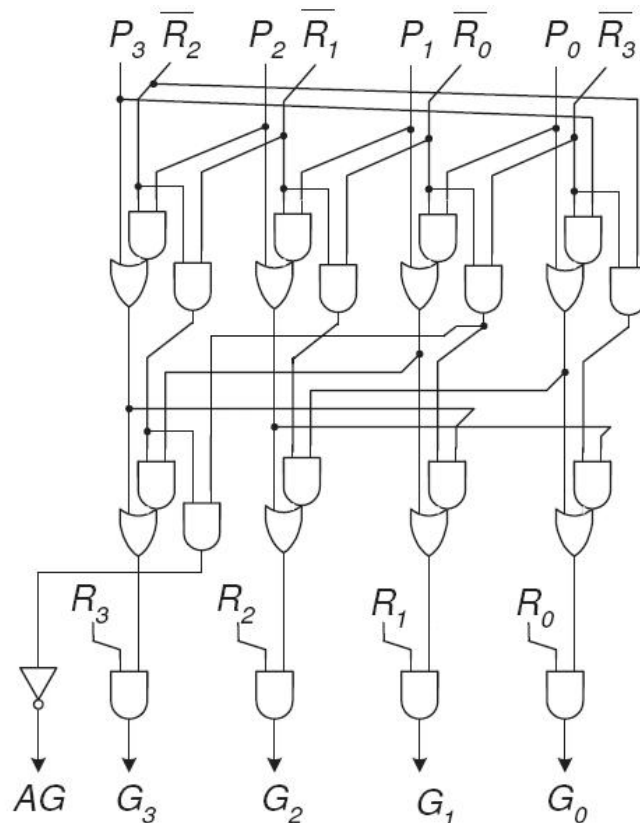
Logical Effort - Ένα ολοκληρωμένο παράδειγμα σχεδίασης

Μανόλης Καλλίγερος (kalliger@aegean.gr)

### Διαδικαστικά

Η καταληκτική ημερομηνία παράδοσης της άσκησης είναι η Κυριακή 12/6/2011. Ο τρόπος παράδοσης της άσκησης θα είναι **ηλεκτρονικός** (αναφορά + τη βιβλιοθήκη που θα έχετε φτιάξει στο Electric).

### Εισαγωγή



Το παραπάνω κύκλωμα είναι ένας arbiter (κύκλωμα διαιτησίας) των 4 bit. Το κύκλωμα αυτό λαμβάνει σαν είσοδο ένα διάνυσμα αιτήσεων ( $R_3$ ,  $R_2$ ,  $R_1$  και  $R_0$ ) μαζί με τις ανεστραμμένες εκδοχές τους, καθώς και ένα διάνυσμα προτεραιοτήτων ( $P_3$ ,  $P_2$ ,  $P_1$  και  $P_0$ ). Μόνο ένα bit του διανύσματος προτεραιοτήτων επιτρέπεται να είναι ενεργοποιημένο κάθε φορά (κωδικοποίηση one-hot). Ανάλογα με τις τιμές των διανυσμάτων αυτών, ο arbiter

επιλέγει μία από τις αιτήσεις και ενεργοποιεί την αντίστοιχη έξοδο έγκρισης αίτησης ( $G_i$  - grant). Για παράδειγμα, αν  $\{P_3, P_2, P_1, P_0\} = \{0, 0, 1, 0\}$  (έχει προτεραιότητα η συσκευή 1) και  $\{R_3, R_2, R_1, R_0\} = \{1, 0, 1, 1\}$  (έχουν κάνει αίτηση οι συσκευές 3, 1 και 0), τότε ο arbiter ενεργοποιεί την έξοδο  $G_1$  (δηλ., εγκρίνει την αίτηση της συσκευής 1, η οποία έχει μεγαλύτερη προτεραιότητα έναντι των υπολοίπων τριών). Σε περίπτωση που δεν έχει κάνει αίτηση η συσκευή με τη μεγαλύτερη προτεραιότητα, τότε επιλέγεται εκείνη με την αμέσως μικρότερη σύμφωνα με την πολιτική round-robin (κυκλική μετατόπιση της προτεραιότητας). Για παράδειγμα, αν και πάλι  $\{P_3, P_2, P_1, P_0\} = \{0, 0, 1, 0\}$ , αλλά  $\{R_3, R_2, R_1, R_0\} = \{0, 1, 0, 1\}$  (έχει προτεραιότητα η συσκευή 1, αλλά αιτήσεις έχουν κάνει μόνο οι συσκευές 2 και 0), τότε εγκρίνεται η αίτηση της συσκευής 2 (δηλ.,  $\{G_3, G_2, G_1, G_0\} = \{0, 1, 0, 0\}$ ), αφού αυτή είναι η συσκευή με την αμέσως μικρότερη προτεραιότητα μετά τη συσκευή 1, σύμφωνα με τη round-robin πολιτική (υποθέτουμε κυκλική μετατόπιση των προτεραιοτήτων από τα δεξιά προς τα αριστερά). Τέλος, αν καμία συσκευή δεν έχει κάνει αίτηση, όλες οι έξοδοι  $G_i$  απενεργοποιούνται (γίνονται ίσες με το λογικό 0), ενώ ταυτόχρονα απενεργοποιείται και η έξοδος  $AG$  (Any Grant). Η έξοδος  $AG$ , όπως υποδηλώνει και το όνομά της, παραμένει ενεργοποιημένη (ίση με το λογικό 1) όσο οποιαδήποτε από τις εξόδους  $G_i$  είναι 1.

Σημειώνεται, ότι arbiters σαν και τον παραπάνω περιλαμβάνονται σε κυκλώματα διακοπών (switches), τα οποία αποτελούν ένα από τα βασικότερα συστατικά των δικτύων διασύνδεσης (interconnection networks) των σύγχρονων, περίπλοκων ολοκληρωμένων συστημάτων (Systems-on-Chip – SoCs).

Στόχος της άσκησης είναι να σχεδιάσετε τον παραπάνω 4-bit arbiter, έχοντας σαν στόχο τη μικρότερη δυνατή καθυστέρηση (άρα, θα πρέπει να χρησιμοποιήσετε την μέθοδο του Logical Effort). Η άσκηση χωρίζεται σε δύο μέρη:

### Μέρος 1ο: Βελτιστοποίηση του κυκλώματος με χρήση της μεθόδου του Logical Effort

Λάβετε υπόψη σας τα ακόλουθα:

- Κάθε είσοδος του κυκλώματος ( $P_i, R_i, \overline{R_i}$ ) μπορεί να οδηγήσει χωρητικότητα ίση με  $14C$ , όπου  $C$  είναι η χωρητικότητα ενός τρανζίστορ ελαχίστου πλάτους (4λ).
- Κάθε έξοδος του κυκλώματος ( $G_i, AG$ ) μπορεί να οδηγήσει χωρητικότητα ίση με  $84C$ .
- Οι πύλες που θα χρησιμοποιηθούν για την υλοποίηση του κυκλώματος θα είναι static CMOS, με τα πλάτη των τρανζίστορ κατάλληλα επιλεγμένα, ώστε το ρεύμα κατά την ανοδική μετάβαση του κόμβου εξόδου να είναι ίσο με αυτό της καθοδικής μετάβασης (και ίσο με το αντίστοιχο ρεύμα ενός μοναδιαίου αντιστροφέα).
- Όσο αφορά το logical effort ( $g$ ) και το parasitic delay ( $p$ ) των διαφόρων πυλών, χρησιμοποιήστε τις θεωρητικές τιμές που υπολογίζονται από τα πλάτη των τρανζίστορ, οι οποίες περιλαμβάνονται στους πίνακες του βιβλίου και των διαφανειών. Αν χρειαστείτε τα  $p$  και  $g$  για κάποια πύλη που δεν περιλαμβάνεται στους πίνακες αυτούς, υπολογίστε τις τιμές τους με βάση, και πάλι, τα πλάτη των τρανζίστορ της πύλης.

Για τη βελτιστοποίηση του κυκλώματος ακολουθήστε τα παρακάτω βήματα:

- Βρείτε όλα τα διαφορετικά μονοπάτια του κυκλώματος. Αν το ίδιο μονοπάτι

επαναλαμβάνεται περισσότερες από μία φορές, μελετήστε το μόνο μία.

- Αναγνωρίστε το μονοπάτι με τη μεγαλύτερη καθυστέρηση (critical path – κρίσιμο μονοπάτι) και υπολογίστε με τη βοήθεια της μεθόδου του Logical Effort την ελάχιστη τιμή της καθυστέρησης αυτής. Εναλλακτικά, μπορείτε να υπολογίσετε την ελάχιστη καθυστέρηση όλων των μονοπατιών του κυκλώματος ώστε να βρείτε αυτό με τη μεγαλύτερη (το/τα μονοπάτι/τια με την καθυστέρηση αυτή θα είναι και το/τα critical), αλλά είναι σχετικά εύκολο να σκεφθείτε ποιο μονοπάτι μπορεί να είναι το πιο αργό.

- Στο παραπάνω βήμα, αλλά και σε όλους τους υπολογισμούς σας, για να βρείτε το branching effort (b) χρησιμοποιήστε τον τύπο  $(C_{\text{onpath}} + C_{\text{offpath}}) / C_{\text{onpath}}$  (μη θέσετε δηλαδή το branching effort ενός κόμβου ίσο με τον αριθμό των κλάδων που οδηγούνται από αυτόν, αλλά χρησιμοποιήστε τον τύπο από τον ορισμό του branching effort – πιο ακριβής υπολογισμός). Για το branching effort των κόμβων που οδηγούνται από τις εισόδους  $P_i$  και  $\overline{R}_i$  (δηλ., πριν τις πύλες του 1ου επιπέδου), θεωρήστε ότι οι σύνθετες πύλες και οι πύλες AND (NAND σε ένα CMOS κύκλωμα) κλιμακώνονται κατά τον ίδιο παράγοντα k σε σχέση με τις αντίστοιχες πύλες «οδηγούς» (template gates). Με άλλα λόγια, αν οι σύνθετες πύλες είναι, για παράδειγμα, 2 φορές μεγαλύτερες από την αντίστοιχη σύνθετη πύλη που παρέχει το ίδιο ρεύμα με το μοναδιαίο αντιστροφέα, το ίδιο συμβαίνει και για τις πύλες NAND (είναι 2 φορές μεγαλύτερες σε σχέση με τη NAND που δίνει το ίδιο ρεύμα με το μοναδιαίο αντιστροφέα). Αυτό μπορεί να εξηγηθεί από τη δομή του κυκλώματος και το είδος των πυλών (μπορείτε αν θέλετε να προσπαθήσετε να το εξηγήσετε, χωρίς αυτό να προσμετράται στη βαθμολογία).

- Από την τιμή του F του κρίσιμου μονοπατιού, υπολογίστε το βέλτιστο αριθμό επιπέδων που πρέπει να έχει το κύκλωμά σας, ώστε να επιτυγχάνει την ελάχιστη δυνατή καθυστέρηση.

- Τροποποιήστε το κύκλωμα σας (δηλ., τις πύλες του, χωρίς φυσικά να αλλάζουν οι συναρτήσεις εξόδου που παράγονται), ώστε το πλήθος των επιπέδων του (στα κρίσιμα μονοπάτια), να συμπίπτει με τον αριθμό των επιπέδων που υπολογίσατε στο προηγούμενο βήμα. Αν θέλετε μπορείτε να δοκιμάσετε διάφορες τοπολογίες με το βέλτιστο αριθμό επιπέδων και να κρατήσετε την ταχύτερη (σημείωση: ο έλεγχος διαφορετικών τοπολογιών δεν είναι απαραίτητος για την επιτυχή ολοκλήρωση της άσκησης – **μία** τοπολογία με το βέλτιστο αριθμό επιπέδων είναι αρκετή).

- Υπολογίστε τη βέλτιστη καθυστέρηση του κρίσιμου μονοπατιού του νέου σας κυκλώματος, καθώς και το βέλτιστο αριθμό επιπέδων από τη νέα τιμή του F, ώστε να διαπιστώσετε ότι συμπίπτει με το βέλτιστο αριθμό επιπέδων που βρήκατε προηγουμένως.

- Υπολογίστε το  $C_{\text{in}}$  (χωρητικότητα εισόδου) των πυλών που βρίσκονται στο κρίσιμο μονοπάτι, και από αυτή βρείτε τα μεγέθη των τρανζίστορ που τις απαρτίζουν.

- Από το προηγούμενο βήμα και από την υπόθεση σχετικά με το μέγεθος των πυλών του 1ου επιπέδου, θα έχετε καταλήξει στο μέγεθος των περισσότερων πυλών του κυκλώματος. Θα σας απομένει μόνο να αποφασίσετε για το μέγεθος της πύλης (2ου επιπέδου) που οδηγεί τον αντιστροφέα που παράγει την έξοδο AG, καθώς και για το μέγεθος του αντιστροφέα αυτού. Όσο αφορά την πύλη του 2ου επιπέδου, μπορείτε να κάνετε μία υπόθεση που θα σας βολεύει (π.χ., ότι κλιμακώνεται κατά τον ίδιο παράγοντα, σε σχέση με τις υπόλοιπες πύλες

2ου επιπέδου, ως προς την αντίστοιχη πύλη «οδηγό» – είστε ελεύθεροι να κάνετε όποια υπόθεση θέλετε για το μέγεθος της πύλης αυτής, εξηγώντας το λόγο για τον οποίο σας βολεύει να κάνετε την υπόθεση αυτή). Έτσι, ο αντιστροφέας που παράγει την έξοδο  $AG$  θα πρέπει να έχει τέτοιο μέγεθος, ώστε η καθυστέρηση των μονοπατιών που καταλήγουν στην έξοδο  $AG$  να μην υπερβαίνει τη βέλτιστη καθυστέρηση του κρίσιμου μονοπατιού του κυκλώματος που βρήκατε σε προηγούμενο βήμα. Υπολογίστε την καθυστέρηση των μονοπατιών που καταλήγουν στην έξοδο  $AG$ , θέστε τη μικρότερη ή ίση από την καθυστέρηση του κρίσιμου μονοπατιού και λύστε την ανισότητα, ώστε να βρείτε το εύρος τιμών για το  $C_{in}$  του αντιστροφέα αυτού. Στη συνέχεια, επιλέξτε όποια τιμή σας βολεύει ή όποια εσείς κρίνετε καλύτερη (π.χ., τη μικρότερη, για μικρότερη επιφάνεια υλοποίησης και κατανάλωση ισχύος).

- Σαν τελευταίο βήμα της εφαρμογής της μεθόδου του Logical Effort, υπολογίστε την καθυστέρηση όλων των υπολοίπων μονοπατιών του κυκλώματος (εκτός δηλαδή από το κρίσιμο, για το οποίο έχετε βρει τη βέλτιστη καθυστέρηση). Αν η επιλογή του κρίσιμου μονοπατιού που κάνατε είναι σωστή, θα πρέπει η καθυστέρηση των υπολοίπων μονοπατιών να είναι μικρότερη ή ίση από αυτή του κρίσιμου. **ΠΡΟΣΟΧΗ: Ο υπολογισμός της καθυστέρησης των μονοπατιών αυτών δεν θα γίνει από τη σχέση  $N \cdot \sqrt{F} + P$  αλλά από τη σχέση  $\sum_{i=1}^N g_i h_i + P$ . Η πρώτη σχέση δίνει τη βέλτιστη καθυστέρηση ενός μονοπατιού ανεξάρτητα από τα μεγέθη των πυλών (τα μεγέθη προσαρμόζονται στη συνέχεια ώστε να επιτυγχάνεται αυτή η βέλτιστη καθυστέρηση), ενώ η δεύτερη σχέση μας δίνει την καθυστέρηση ενός μονοπατιού με γνωστά μεγέθη πυλών.**

## Μέρος 2ο: Σχεδιασμός του κυκλώματος στο Electric σε επίπεδο layout

Για το σχεδιασμό του κυκλώματος θα χρησιμοποιήσετε το εργαλείο Electric, όπως και στο 1ο Σετ ασκήσεων. Ο σχεδιασμός θα γίνει σε επίπεδο layout, ενώ θα πρέπει να ακολουθήσετε την αρχή της ιεραρχικής σχεδίασης, δηλαδή αρχικά θα σχεδιάσετε όλες τις πύλες που σας είναι απαραίτητες σαν ανεξάρτητα cells (χρησιμοποιώντας τα μεγέθη των τρανζίστορ που υπολογίσατε στο 1ο μέρος της άσκησης) και στη συνέχεια θα χρησιμοποιήσετε τα cells αυτά ώστε να υλοποιήσετε τον ζητούμενο arbiter. Δώστε προσοχή στις ακόλουθες οδηγίες:

- Υλοποιήστε τις πύλες σας σαν standard cells. Αυτό σημαίνει ότι οι πύλες σας θα πρέπει να «κουμπώνουν» όταν τοποθετηθούν η μία δίπλα στην άλλη και, άρα, θα πρέπει να έχουν το ίδιο ύψος. Το ύψος μίας πύλης καθορίζεται από την απόσταση μεταξύ των γραμμών του  $V_{DD}$  και του Gnd. Φροντίστε να φτιάξετε τη μεγαλύτερη σας πύλη (σε ύψος) πρώτη, ώστε στις υπόλοιπες να διατηρήσετε την ίδια απόσταση μεταξύ των γραμμών τροφοδοσίας και γείωσης.
- Οι γραμμές τροφοδοσίας και γείωσης των πυλών θα πρέπει να είναι οριζόντιες, μετάλλου-1, με πλάτος 8λ. Όλες οι υπόλοιπες γραμμές μετάλλου του κυκλώματος θα πρέπει να έχουν πλάτος 4λ. Συνολικά, για την υλοποίηση ολόκληρου του κυκλώματος επιτρέπεται

να χρησιμοποιήσετε γραμμές μετάλλου-1 και μετάλλου-2. Στις πύλες σας χρησιμοποιήστε μόνο γραμμές μετάλλου-1 και πολυσιλικόνης (για τις συνδέσεις των gates των τρανζίστορ).

- Αν σε κάποιες πύλες πρέπει να χρησιμοποιήσετε τρανζίστορ πολύ μεγάλου πλάτους, εφαρμόστε την τεχνική του folding (σελ. 556 του βιβλίου). Έτσι θα αποφύγετε το υπερβολικό ύψος των πυλών, εξαιτίας του μεγέθους των τρανζίστορ ενός συγκεκριμένου είδους πυλών.

- Φροντίστε κάθε πύλη σας να έχει ένα ελάχιστο περιθώριο 2λ σε κάθε πλευρά της, σε σχέση με την τελευταία γραμμή μετάλλου-1 σε εκείνη την πλευρά, έτσι ώστε αν τοποθετηθούν δύο πύλες η μία δίπλα στην άλλη ή η μία κάτω από την άλλη, να μην παραβιάζεται η ελάχιστη απόσταση μεταξύ γειτονικών γραμμών μετάλλου (4λ).

- Φροντίστε στις εισόδους και στην έξοδο κάθε πύλης να υπάρχει μία νία προς το επίπεδο μετάλλου 2 (για να συνδέσετε πολυσιλικόνη με μέταλλο-2 θα πρέπει να τοποθετήσετε, τη μία πάνω στην άλλη, μία νία πολυσιλικόνης-μετάλλου-1 και μία νία μετάλλου-1-μετάλλου-2 – stacked vias). Δεν είναι υποχρεωτικό όλες οι συνδέσεις στις εισόδους και εξόδους των πυλών να είναι με γραμμές μετάλλου-2 (μπορείτε να χρησιμοποιήσετε και μέταλλο-1 αν σας βολεύει). Παρόλα αυτά, η ύπαρξη των νιων προς το επίπεδο μετάλλου 2, σας δίνει τη δυνατότητα να χρησιμοποιήσετε όποιο είδος γραμμής θέλετε (μέταλλο-1 ή μέταλλο-2) για τη σύνδεση των εισόδων και εξόδων των πυλών. Όταν τελικά αποφασίσετε πως θα κάνετε τη διασύνδεση κάθε επιπέδου πυλών του arbiter με το επόμενο, μπορείτε (αν θέλετε) να αφαιρέσετε από τα cells σας όσες νιες προς μέταλλο-2 δεν πρόκειται να χρησιμοποιηθούν (σε όσες δηλαδή εισόδους ή εξόδους θα χρησιμοποιηθεί μέταλλο-1 για τη σύνδεσή τους με τις γειτονικές πύλες).

- Προσπαθήστε οι εισόδοι και οι εξόδοι των πυλών να έχουν μεταξύ τους απόσταση 4λ, είτε κατά τον οριζόντιο, είτε κατά τον κατακόρυφο άξονα (ή, αν είναι δυνατόν, και κατά τους δύο άξονες). Αυτό θα σας βοηθήσει στη φάση της διασύνδεσης των πυλών, αφού θα υπάρχει η κατάλληλη απόσταση ώστε να μπορούν παράλληλα καλώδια ενός επιπέδου μετάλλου να συνδεθούν με τις εισόδους και εξόδους των πυλών.

- Περισσότερες πληροφορίες σχετικά με τη σχεδίαση πυλών σαν standard cells μπορείτε να βρείτε στην παράγραφο XI του αρχείου Lab0.pdf (βλ. 1ο Σετ Ασκήσεων), καθώς και στην Παράγραφο 8.8.2 του βιβλίου (σελ. 553-556).

- Όσο αφορά τη διασύνδεση των cells που φτιάξατε με στόχο την υλοποίηση του ζητούμενου arbiter, τοποθετήστε τις πύλες σας με τέτοιο τρόπο ώστε να σχηματιστεί ένα δισδιάστατο array (διάταξη ορθογωνίου παραλληλογράμμου).

- Για τη διασύνδεση των πυλών μπορείτε να χρησιμοποιήσετε γραμμές μετάλλου-1 και μετάλλου-2 (αλλά όχι μετάλλου-3). Σε μεγαλύτερους σχεδιασμούς, συνήθως περιορίζεται η χρήση των γραμμών ενός επιπέδου μετάλλου, έτσι ώστε να σχεδιάζονται όλες κατά μία συγκεκριμένη κατεύθυνση (π.χ., το μέταλλο-2 κατακόρυφα, το μέταλλο-3 οριζόντια, κ.ο.κ.). Για το ζητούμενο κύκλωμα δεν υπάρχει τόσο αυστηρός περιορισμός. Παρόλα αυτά, θα πρέπει να τηρήσετε κάποιον κανόνα για τις γραμμές που τέμνονται. Και αφού οι γραμμές τροφοδοσίας και γείωσης είναι οριζόντιες μετάλλου-1, φροντίστε, όπου χρειαστείτε κατακόρυφες γραμμές που τέμνονται με οριζόντιες, αυτές να είναι σε μέταλλο-2. Γενικά

προσπαθήστε οι οριζόντιες γραμμές του κυκλώματός σας να είναι σε μέταλλο-1 και οι κατακόρυφες σε μέταλλο-2 (ειδικά αυτές που έχουν μεγάλο μήκος). Αν τώρα κάπου χρειαστείτε μία μικρή γραμμή ή αν μία γραμμή πρέπει σε κάποιο σημείο της να αλλάξει κατεύθυνση, μπορείτε, αν δεν σας δημιουργεί πρόβλημα, να χρησιμοποιήσετε όποιο επίπεδο μετάλλου θέλετε ή να συνεχίσετε να χρησιμοποιείτε το ίδιο επίπεδο μετάλλου, αν πρόκειται για γραμμή που αλλάζει κατεύθυνση.

- Συνδέστε μεταξύ τους τις γραμμές  $V_{DD}$  και Gnd όλων των πυλών που βρίσκονται στην ίδια γραμμή του δισδιάστατου array που φτιάξατε και, στη συνέχεια, δημιουργήστε ένα νέο Export για κάθε ενιαία γραμμή  $V_{DD}$  και Gnd. Φυσικά, νέα Exports θα χρειαστεί να δημιουργήσετε και για όλες τις εισόδους και εξόδους του τελικού σας κυκλώματος.

- Περισσότερες πληροφορίες σχετικά με την ιεραρχική σχεδίαση στο Electric μπορείτε να βρείτε στην παράγραφο XV του αρχείου Lab0.pdf.