

1

ΕΤΕΡΟΓΕΝΗ ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

FIELD PROGRAMMABLE GATE ARRAYS (FPGA)

Επικ.Καθηγητής Μιχάλης Ψαράκης

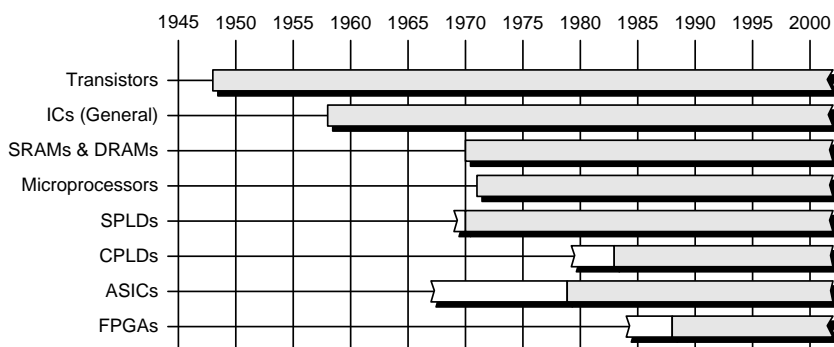
Ενότητα 1

2

- Ιστορική αναδρομή στις προγραμματιζόμενες συσκευές λογικής
- Μια ματιά στους προγόνους των FPGAs

Προέλευση των FPGAs

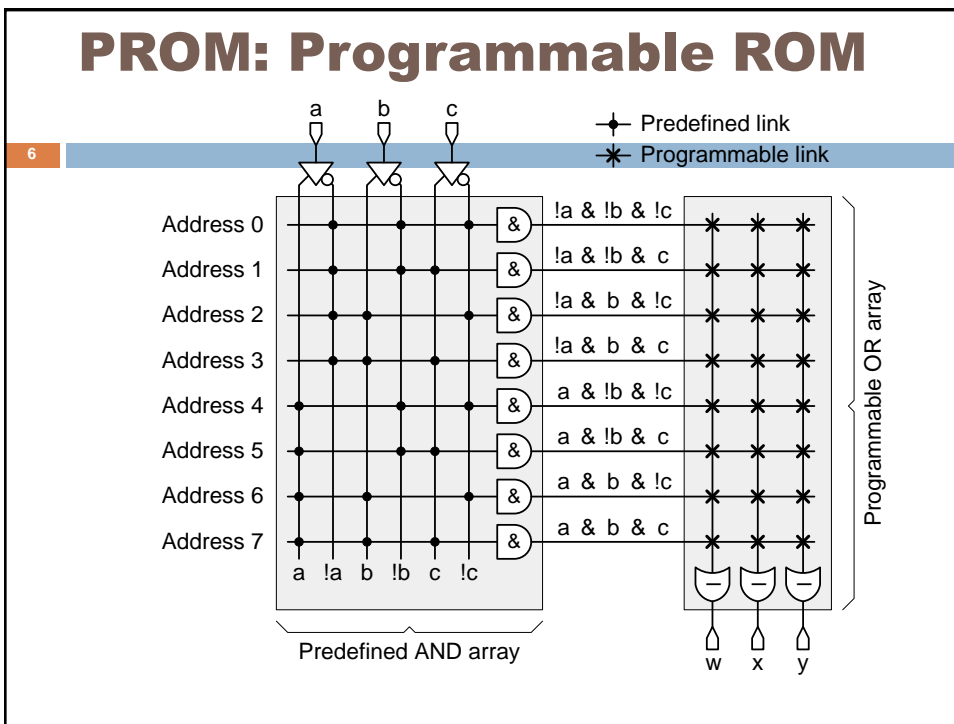
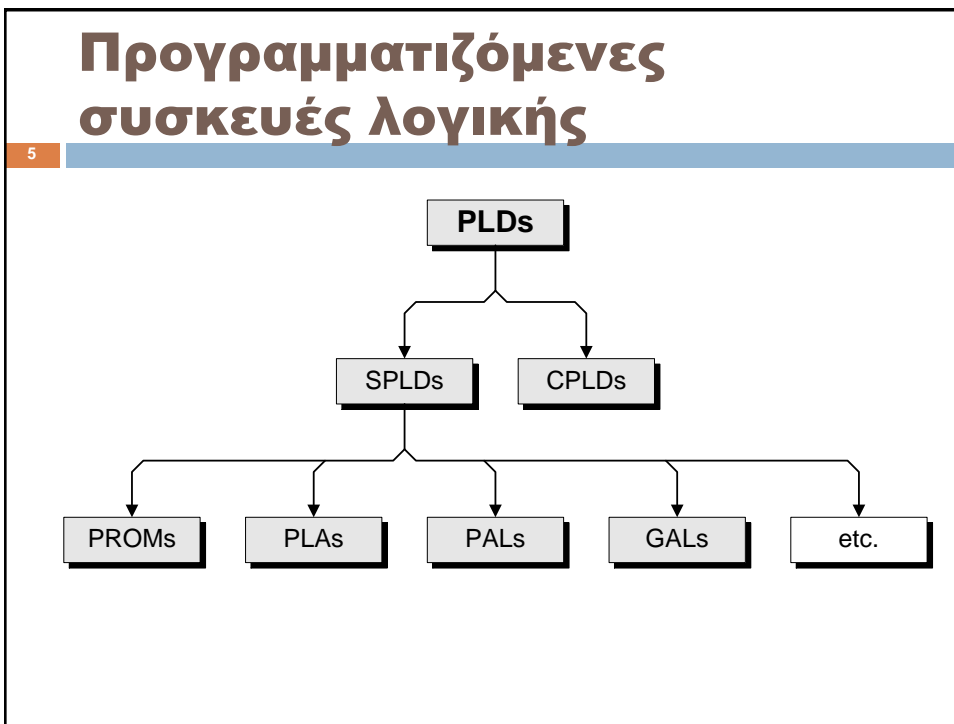
3



Ολοκληρωμένα κυκλώματα

4

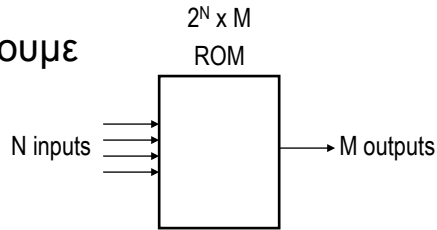
- **Small, medium and large scale integration**
 - **SSI, MSI**
 - Έως 200 πύλες ανά συσκευή
 - Πιο κοινές συσκευές: 74xx
 - Πύλες, flip flops, latches, καταχωρητές, μετρητές, κτλ.
 - **LSI**
 - Από 200 έως 200,000 πύλες ανά συσκευή
 - Μικρές μνήμες, προγραμματιζόμενες συσκευές λογικής
- **Very and ultra large scale integration**
 - **VLSI**
 - Πάνω από 200,000 πύλες ανά συσκευή
 - Συνήθως, ο αριθμός πυλών αντικαθίσταται από αριθμό τρανζίστορ
 - **ULSI**
 - Εκατομμύρια τρανζίστορ
 - SoC (System-on-Chip), 3D-IC (3 dimensional IC)



PROM

7

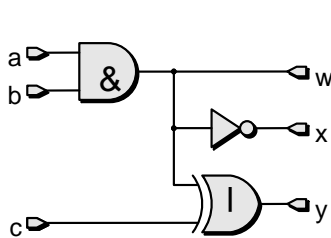
- Μπορούμε να υλοποιήσουμε M συναρτήσεις των N εισόδων σε αυτήν την PROM



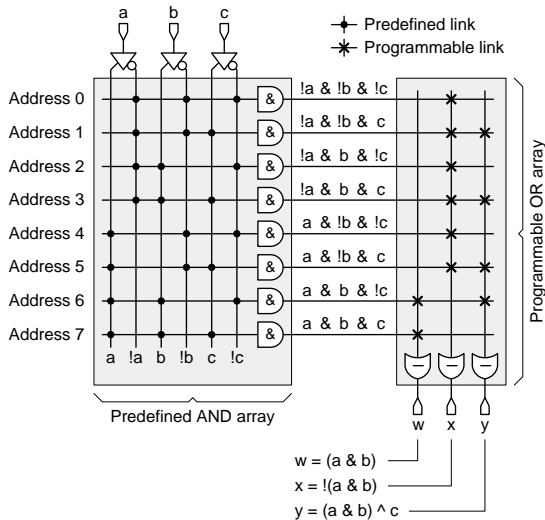
- ▣ Καθορίζουμε τον πίνακα αληθείας των συναρτήσεων που θέλουμε να προγραμματίσουμε
- ▣ Δεν υπάρχει λόγος να απλοποιήσουμε τις συναρτήσεις

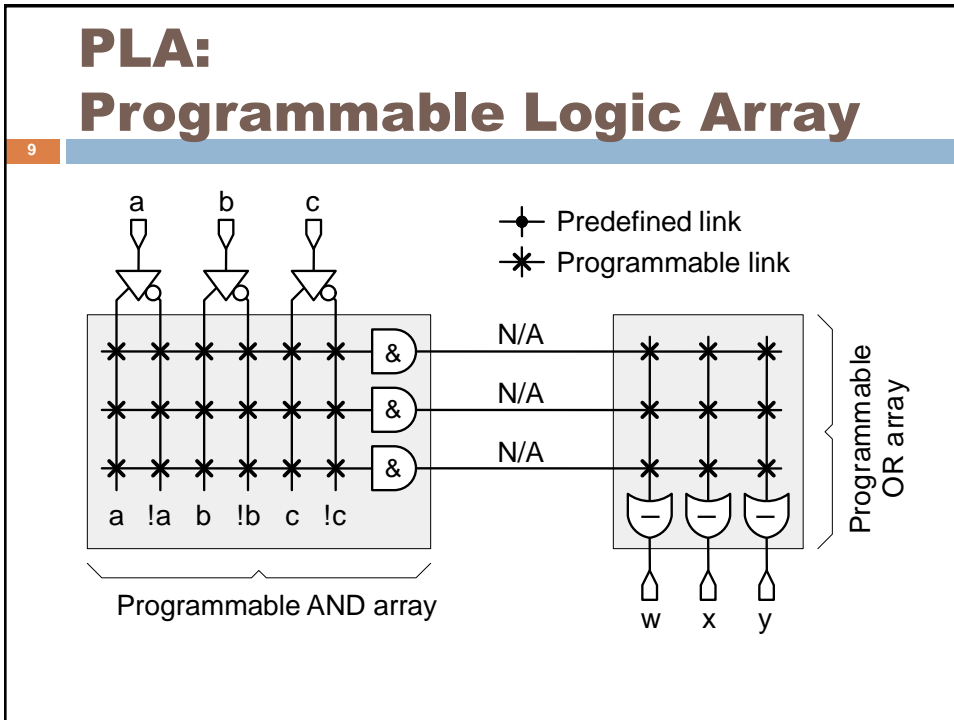
PROM: παράδειγμα

8

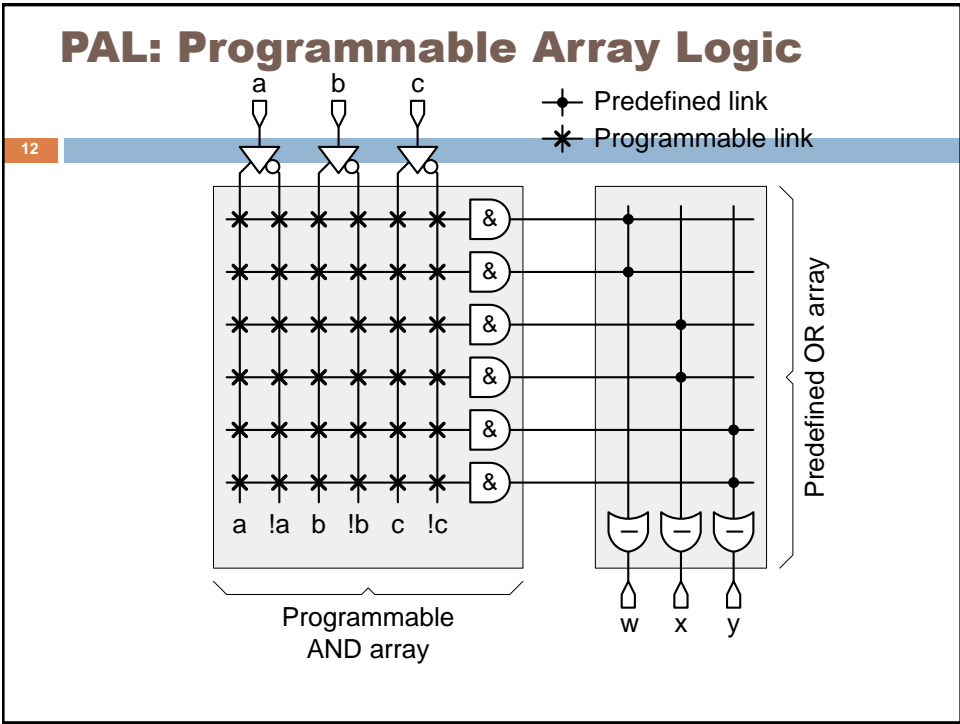
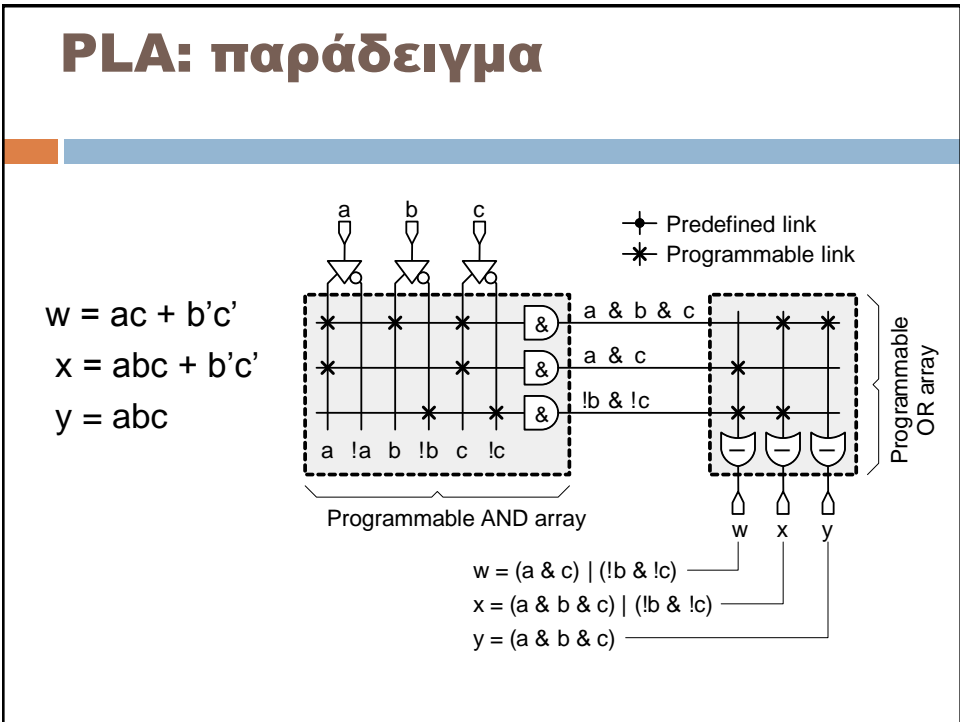


a	b	c	w	x	y
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	0	0





- ## PLA
- 10
- Ο αριθμός των συναρτήσεων AND είναι **ανεξάρτητος** από τον αριθμό των εισόδων
 - Μπορούμε να υλοποιήσουμε συναρτήσεις χρησιμοποιώντας τους διαθέσιμους ελαχιστόρους (**minterms**), οι οποίοι μοιράζονται μεταξύ των συναρτήσεων (**product term sharing**)
 - ▣ Πρέπει να μειώσουμε τις συναρτήσεις σε άθροισμα γινομένων, ελπίζοντας ότι μπορούν να υλοποιηθούν με τους διαθέσιμους ελαχιστόρους
 - ▣ CAD (computer-aided design) εργαλεία εκτελούν τη βελτιστοποίηση



PAL

13

- Μπορούμε να υλοποιήσουμε συναρτήσεις χρησιμοποιώντας τους διαθέσιμους ελαχιστόρους (minterms)
 - Δεν επιτρέπεται το μοίρασμα των ελαχιστόρων μεταξύ των συναρτήσεων (**no product term sharing**)
 - Πάλι, οι συναρτήσεις πρέπει να βελτιστοποιηθούν, εάν είναι εφικτό
- Τα PALs είναι **γρηγορότερα** από τα PLAs

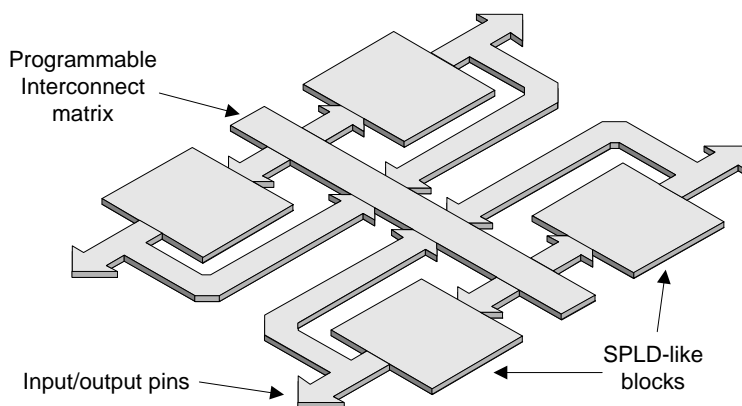
Πρόσθετες επιλογές προγραμματισμού

14

- Πρόσθετες επιλογές προγραμματισμού που παρέχονται σε συσκευές PLAs και PALs:
 - Αντιστροφή των εξόδων
 - Διαμόρφωση ενός ακροδέκτη σαν είσοδο ή έξοδο
 - Tristate outputs
 - Registered outputs

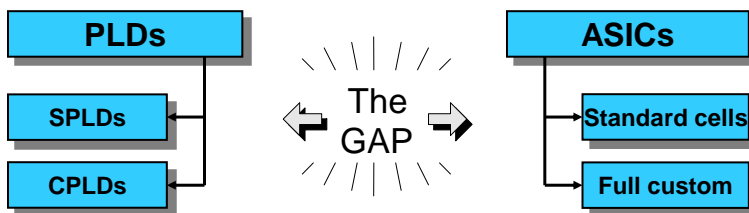
CPLDs

- CPLD = Complex Programmable Logic Device



Το χάσμα

16

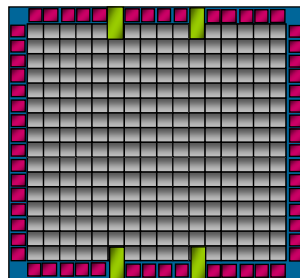


- Για να λύσει το πρόβλημα, η εταιρεία Xilinx ανέπτυξε μια νέα κλάση ICs, τα FPGAs (field-programmable gate arrays), τα οποία βγήκαν στην αγορά το 1984

FPGA: Field Programmable Gate Array

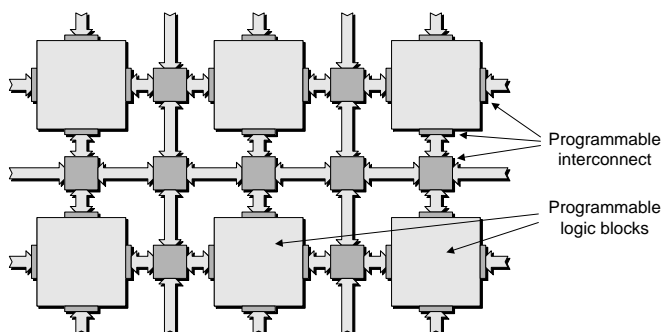
17

- Τι είναι ένα FPGA; Όλα τα είδη υλικού ...
 - ▣ I/O Cells
 - ▣ Logic Cells
 - ▣ Memories
 - ▣ Microprocessors
 - ▣ Clock Management
 - ▣ High Speed I/O Transceivers
 - ▣ Programmable routing



FPGAs

- Προγραμματιζόμενες συσκευές λογικής
 - ▣ Λογικά μπλοκ (logic blocks)
 - ▣ Διασύνδεση (interconnect)



Ενότητα 2

19

- Πλεονεκτήματα της χρήσης των προγραμματιζόμενων συσκευών λογικής
- Γιατί θα προτιμήσουμε να χρησιμοποιήσουμε ένα FPGA για να υλοποιήσουμε το σύστημά μας και όχι ένα ASIC;

Κόστος παραγωγής των ICs

20

- Μία τυπική γραμμή παραγωγής ICs:
 - καταλαμβάνει περίπου ένα οικοδομικό τετράγωνο
 - απασχολεί μερικές εκατοντάδες υπαλλήλους
- Ο χρόνος εξυπηρέτησης μίας νέας διαδικασίας παραγωγής διαρκεί 6-8 μήνες
- Η πιο προσοδοφόρα περίοδος του προϊόντος είναι οι πρώτοι 18-24 μήνες

Παράγοντες κόστους

21

- **Για μεγάλες ποσότητες:**
 - ▣ συσκευασία (packaging), το μεγαλύτερο κόστος
 - ▣ έλεγχος ορθής λειτουργίας (testing), το δεύτερο μεγαλύτερο κόστος
- **Για μικρές ποσότητες,** το κόστος σχεδίασης υπερκαλύπτει όλα τα κόστη παραγωγής

FPGAs & ASICs

22

- **Field Programmable Gate Arrays (FPGAs) --
πρότυπα ολοκληρωμένα κυκλώματα**
 - ▣ προ-κατασκευασμένα
 - ▣ ο σχεδιαστής χρησιμοποιεί το υπάρχον τσιπ για να υλοποιήσει την εφαρμογή
- **Application Specific Integrated Circuits (ASICs) --
Ολοκληρωμένα κυκλώματα ειδικής εφαρμογής**
 - ▣ προσαρμόζονται στην εφαρμογή
 - ▣ ο σχεδιαστής πρέπει να ασχοληθεί με τη φυσική σχεδίαση
 - ▣ η εταιρεία πρέπει να κατασκευάσει το τσιπ

FPGAs ή ASICs;

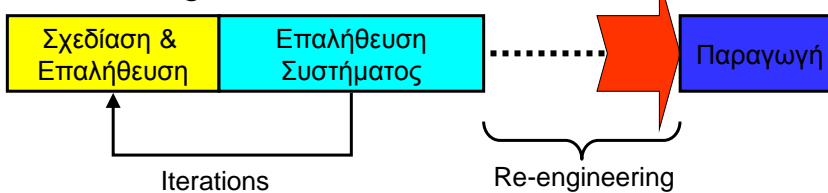
23

- Θα σχεδιάσετε το σύστημά σας με τη χρήση FPGA ή ASIC;
 - ▣ Τα FPGAs έχουν μικρότερο κύκλο σχεδίασης
 - ▣ Τα FPGAs δεν απαιτούν χρόνο κατασκευής
 - ▣ Τα FPGAs διευκολύνουν τη διαδικασία δοκιμής και διόρθωσης του πρωτοτύπου
 - ▣ Τα FPGAs είναι πιο αργά, πιο μεγάλα και καταναλώνουν περισσότερη ενέργεια
 - Διαβάστε το άρθρο :
 - ▣ Kuon, I.; Rose, J.; , "Measuring the Gap Between FPGAs and ASICs," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Feb. 2007

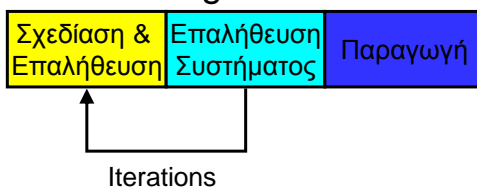
FPGAs εναντίον ASICs

24

ASIC Design Flow



FPGA Design Flow



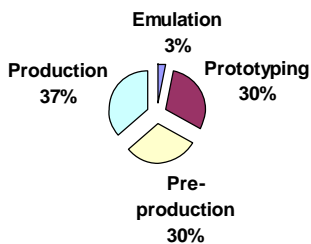
Γιατί FPGAs;

25

- **Λογική συγκόλλησης (glue logic)**
 - ▣ Αντικαθιστούν τα SSI & MSI τσιπ
 - ▣ Μειώνουν τον κατάλογο υλικών
- **Μειώνουν το ρίσκο (low risk)**
 - ▣ Γρήγορη αλλαγή της συνάρτησης (rapid turnaround)
- **Εξομοίωση (emulation)**
- **Δυναμικά επαναπροσδιοριζόμενο υλικό (dynamically reconfigurable hardware)**

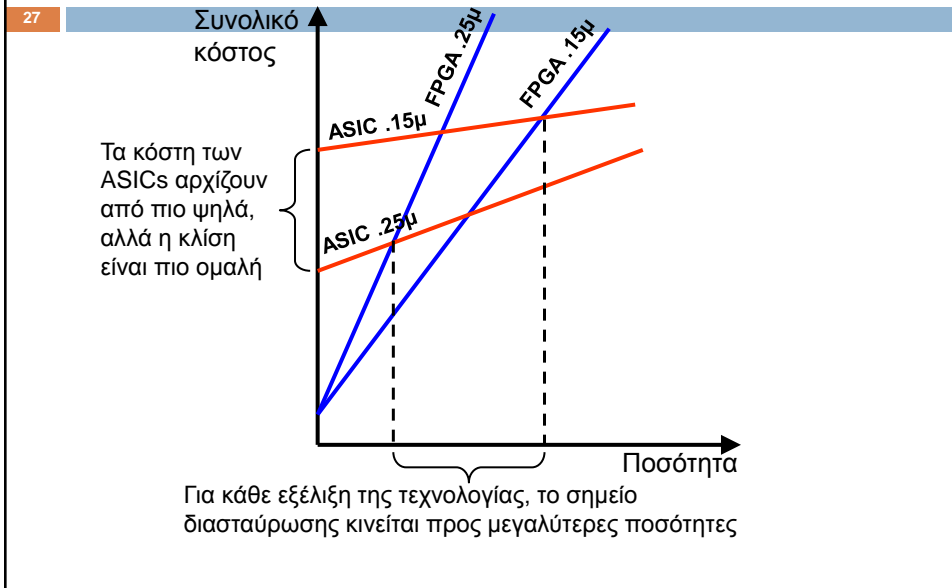
Που χρησιμοποιούνται τα FPGAs;

26



	Χρόνος στην αγορά	Απόδοση	Ποσότητα
Emulation	Σημαντικός	Δεν ενδιαφέρει	Χαμηλή
Prototyping	Σημαντικός	Δεν ενδιαφέρει	Χαμηλή
Pre-production	Σημαντικός	Κρίσιμος	Μέτρια
Production	Σημαντικός	Κρίσιμος	Μεγάλη

Ανάλυση του κόστους



Σχεδίαση ψηφιακών συστημάτων: διαδικασία

28

- Κυριότερα επίπεδα αφαίρεσης της σχεδίασης ενός ψηφιακού συστήματος
 - ▣ προδιαγραφές (specification)
 - ▣ αρχιτεκτονική (architecture)
 - ▣ λογική σχεδίαση (logic design)
 - ▣ σχεδίαση κυκλώματος (circuit design)
 - ▣ φυσική διάταξη (layout)
- Σχεδίαση με τη χρήση FPGA

Σχεδίαση FPGA

29

- Ο κατασκευαστής FPGA σχεδιάζει μία δομή FPGA (FPGA fabric)
 - ▣ Ο σχεδιαστής χρησιμοποιεί αυτή τη δομή
- Ο κατασκευαστής FPGA πρέπει να λάβει υπόψιν του κάποια σχεδιαστικά θέματα:
 - ▣ Να μελετήσει αντιπροσωπευτικές σχεδιάσεις
 - ▣ Να επιλέξει τοπολογία διασύνδεσης
 - ▣ Να δημιουργήσει δομές λογικών στοιχείων
 - ▣ Να σχεδιάσει κυκλώματα και διάταξη

Πρέπει να ασχοληθούμε με τη φυσική διάταξη των FPGAs;

30

- Δεν σχεδιάζουμε τη φυσική διάταξη
- Όμως η φυσική διάταξη καθορίζει:
 - ▣ την καθυστέρηση της λογικής
 - ▣ την καθυστέρηση των διασυνδέσεων
 - ▣ την κατανάλωση ενέργειας
- Πρέπει να κατανοήσουμε τα χαρακτηριστικά της αρχιτεκτονικής του FPGA:
 - ▣ λογικά στοιχεία
 - ▣ διασύνδεση
 - ▣ επιπλέον δυνατότητες προγραμματιζόμενου υλικού

Ενότητα 3

31

- Βασικές έννοιες: τεχνολογίες προγραμματισμού
- Πλεονεκτήματα & μειονεκτήματα των τεχνολογιών προγραμματισμού

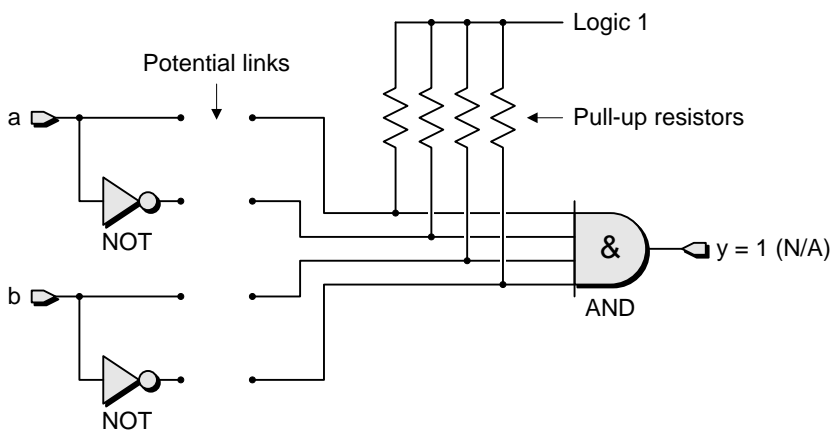
Τεχνολογίες προγραμματισμού

32

- **Antifuse**
 - ▣ Μη-πτητική μνήμη (non-volatile)
 - ▣ Προγραμματίζεται μία φορά (one-time programmable, OTP)
- **Flash**
 - ▣ Μη-πτητική μνήμη (non-volatile)
 - ▣ Επανα-προγραμματίζεται (re-programmable)
- **SRAM**
 - ▣ Πτητική μνήμη (volatile)

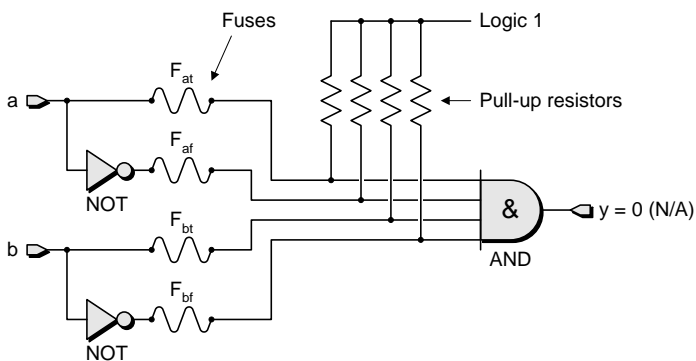
Μια απλή προγραμματιζόμενη συνάρτηση

33



Τεχνολογία εύτηκτων συνδέσεων (fusable-link)

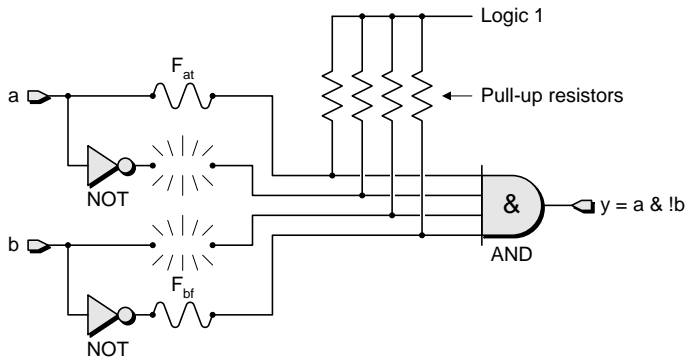
34



- Αρχικά όλες οι **ασφάλειες (fuses)** κάνουν επαφή
- Μπορούμε να απομακρύνουμε επιλεκτικά κάποιες ασφάλειες, εφαρμόζοντας παλμούς υψηλής τάσης και ρεύματος

Programmed fusible links

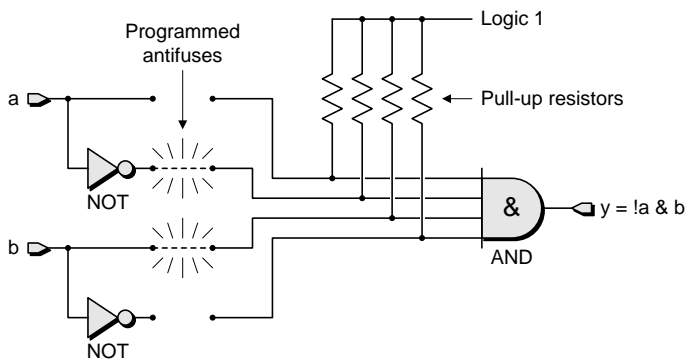
35



- Η διαδικασία ονομάζεται **προγραμματισμός (programming)** ή **κάψιμο (blowing, burning)** της συσκευής
- Οι συσκευές αυτής της τεχνολογίας προγραμματίζονται μόνο μία φορά (**one-time programmable, OTP**)

Τεχνολογίας αντι-ασφάλειας (antifuse)

36

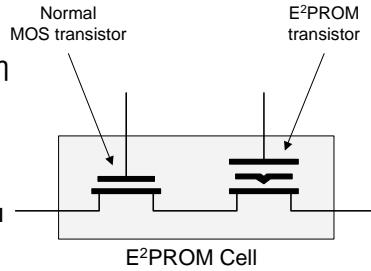


- Η τεχνολογία **antifuse** είναι η αντιδιαμετρική εναλλακτική λύση της τεχνολογίας fuse
- Μία αντι-ασφάλεια (antifuse) έχει υψηλή αντίσταση, ώστε μπορεί να θεωρηθεί ως **ανοικτο-κύκλωμα (open circuit)**

Τεχνολογία EEPROM

37

- Στην τεχνολογία EEPROM η διαγραφή γίνεται με ηλεκτρικό τρόπο
 - Μικρότερος χρόνος διαγραφής
- EEPROM κυψέλη
 - Μεγαλύτερη από μία EPROM κυψέλη
 - Περιέχει 2 transistors
 - Τα επίπεδα μόνωσης του EEPROM transistor είναι πιο λεπτά
 - Το δεύτερο transistor χρησιμοποιείται για να διαγράψει ηλεκτρικά την κυψέλη



Τεχνολογία Flash

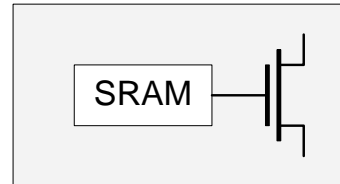
38

- Μικρός χρόνος διαγραφής σε σύγκριση με την τεχνολογία EPROM
- Διαφορετικές flash αρχιτεκτονικές
 - Κυψέλη μονού transistor (όπως η EPROM)
 - Διαγραφή μεγάλων τμημάτων ή ολόκληρης της συσκευής
 - Κυψέλη δυο transistor (όπως η EEPROM)
 - Διαγραφή και επαναπρογραμματισμός σε επίπεδο λέξης

Τεχνολογία SRAM

39

- Πτητική μνήμη
- Η κυψέλη περιέχει
 - Στοιχείο μνήμης SRAM με πολλά transistor (4 ή 6)
 - Transistor ελέγχου
 - Ανάλογα με την τιμή του στοιχείου μνήμης το transistor είναι ON ή OFF
- Μια SRAM κυψέλη είναι μεγαλύτερη από μία EPROM ή flash κυψέλη



Τεχνολογίες προγραμματισμού

40

Τεχνολογία	Σύμβολο	Κυρίως σχετίζεται με ...
Fusible-link		SPLDs
Antifuse		FPGAs
EPROM		SPLDs και CPLDs
E ² PROM/ FLASH		SPLDs και CPLDs (μερικά FPGAs)
SRAM		FPGAs (μερικά CPLDs)

Σύγκριση τεχνολογιών

41

Στοιχείο	SRAM	Antifuse	E2PROM / FLASH
Τεχνολογία	State-of-the-art	Μία ή περισσότερες γενιές πίσω	Μία ή περισσότερες γενιές πίσω
Επαναπρογραμματιζόμενο	Ναι (in system)	Όχι	Ναι (in-system ή offline)
Ταχύτητα προγραμματισμού (συν διαγραφή)	Γρήγορη	---	3x πιο αργή από την SRAM
Πτητική (προγραμματίζεται στο power-up)	Ναι	Όχι	Όχι (αλλά είναι εφικτό)
Απαιτεί εξωτερικό αρχείο	Ναι	Όχι	Όχι
Ιδανικό για πρωτότυπα	Ναι (πολύ καλό)	Όχι	Ναι (μέτρια λύση)
Άμεση ενεργοποίηση	Όχι	Ναι	Ναι
Ασφάλεια IP	Αποδεκτή (ειδικά όταν χρησιμοποιείται κρυπτογράφηση bitstream)	Πολύ καλή	Πολύ καλή
Μέγεθος κυψέλης διαμόρφωσης	Μεγάλο (έξ transistors)	Πολύ μικρό	Μέτριο-μικρό (δύο transistors)
Κατανάλωση ισχύος	Υψηλή	Χαμηλή	Μέτρια
Ανεκτικά στην ακτινοβολία	Όχι	Ναι	Όχι ιδιαίτερα

42

Ενότητα 4

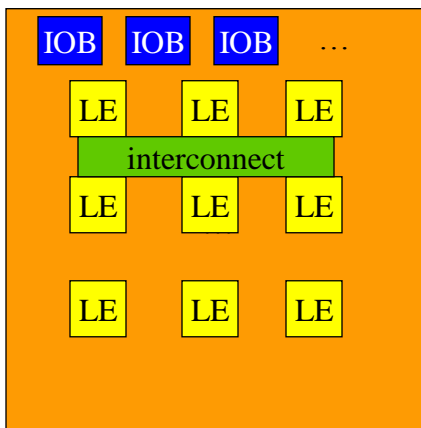
43

- Δομή των FPGAs
- Ποια είναι τα συστατικά στοιχεία των FPGAs;

Βασικά στοιχεία μίας δομής FPGA

44

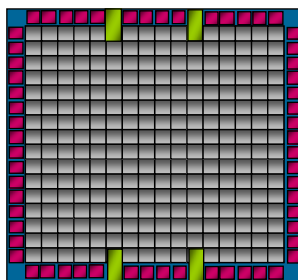
- Λογικά στοιχεία (logic elements)
- Διασύνδεση (interconnect)
- Μπλοκ εισόδου/εξόδου (I/O blocks)



FPGA fabric

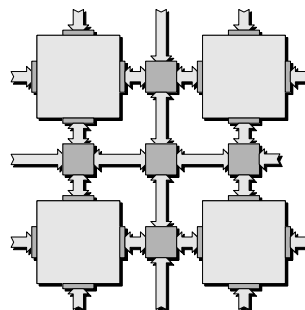
45

- Η δομή ενός FPGA μας παρέχει πολλές δυνατότητες ...
 - ▣ Logic Cells
 - ▣ Programmable routing
 - ▣ I/O Cells
 - ▣ Memories
 - ▣ Microprocessors
 - ▣ Clock Management
 - ▣ High Speed I/O Transceivers



Λογικά στοιχεία

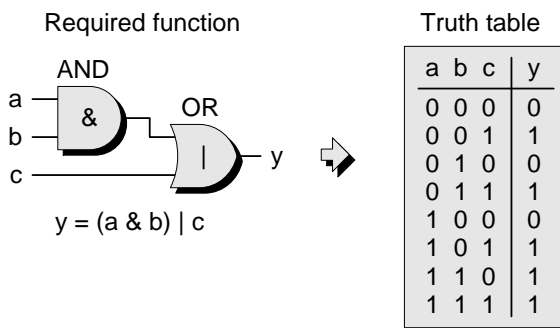
- Περιλαμβάνουν συνδυαστική συνάρτηση και καταχωρητές
- Χρησιμοποιούν μνήμη SRAM σαν πίνακες αναζήτησης (lookup table, LUT) για τη συνδυαστική συνάρτηση
- Μικρότερο επίπεδο λεπτομέρειας σε σύγκριση με τις λογικές πύλες
- Παρέχουν επιπλέον εξειδικευμένη λογική



Παράδειγμα λογικού στοιχείου

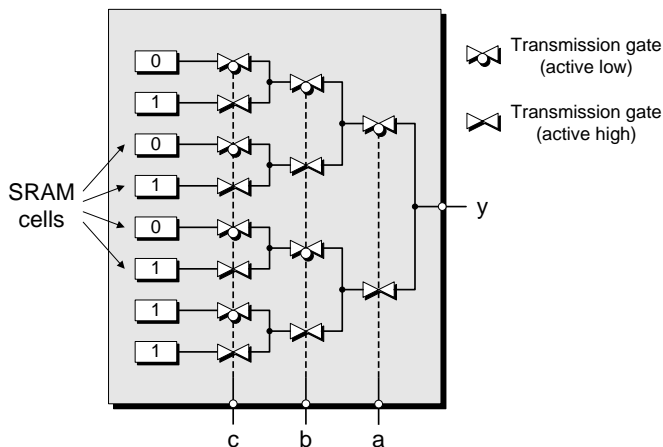
47

- **LUT-based logic element**
 - Υλοποίηση λογικού στοιχείου με χρήση πίνακα αναζήτησης (**LookUp Table, LUT**)



Transmission gate-based LUT

48



Πόσες εισόδους πρέπει να έχει το LUT;

49

- Ένα LUT n -εισόδων μπορεί να υλοποιήσει όλες τις πιθανές συνδυαστικές συναρτήσεις των n -εισόδων
- Προσθέτοντας μία ακόμα είσοδο στο LUT
 - ▣ Είναι δυνατή η αναπαράσταση πιο σύνθετων συναρτήσεων
 - ▣ Διπλασιάζουμε το μέγεθος της SRAM
- Μελέτες έχουν δείξει ότι τα LUT 4-εισόδων είναι μια «καλή» λύση

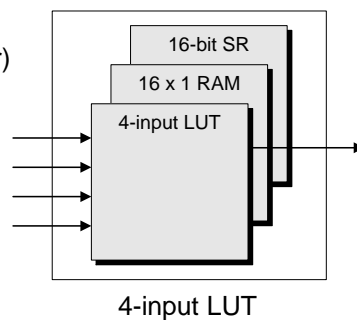
Αξιολόγηση SRAM-based LUT

50

- Όλες οι συναρτήσεις καταλαμβάνουν τον ίδιο χώρο
- Όλες οι συναρτήσεις έχουν την ίδια καθυστέρηση
- Η μνήμη SRAM είναι μεγαλύτερη από το ισοδύναμο κύκλωμα πυλών
- Καταναλώνει ενέργεια σε κατάσταση αδράνειας (idle state)

Πολλαπλές λειτουργίες του LUT

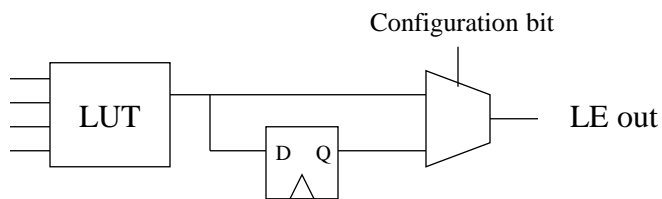
- Εκτός από τη βασική τους λειτουργία, κάποιοι κατασκευαστές παρέχουν τη δυνατότητα στα LUTs να λειτουργήσουν ως:
 - ▣ Μικρό μπλοκ RAM
 - ▣ Καταχωρητής ολίσθησης (shift register)
- Για παράδειγμα ένα LUT 4-εισόδων μπορεί να λειτουργήσει ως
 - ▣ 16×1 RAM
 - ▣ 16-bit καταχωρητής ολίσθησης



Καταχωρητές στα λογικά στοιχεία

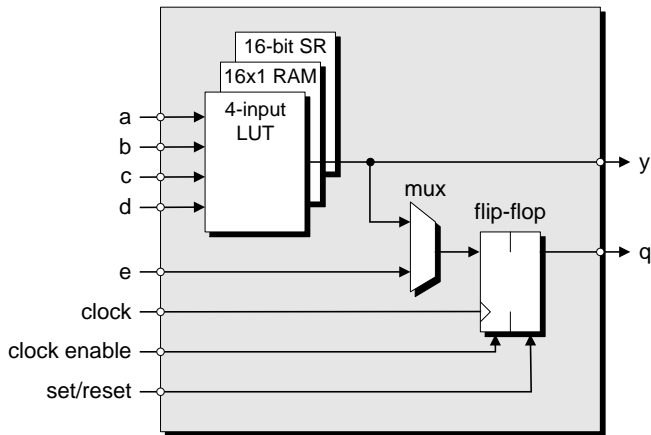
52

- Μπορεί να επιλεγεί καταχωρητής στην έξοδο του λογικού στοιχείου:



Xilinx logic cell (LC)

53

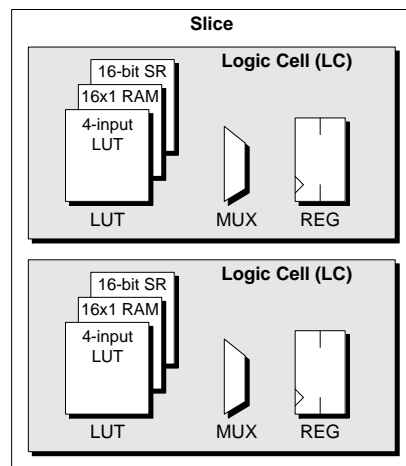


Απλοποιημένη μορφή ενός λογικού στοιχείου της Xilinx

Xilinx slice

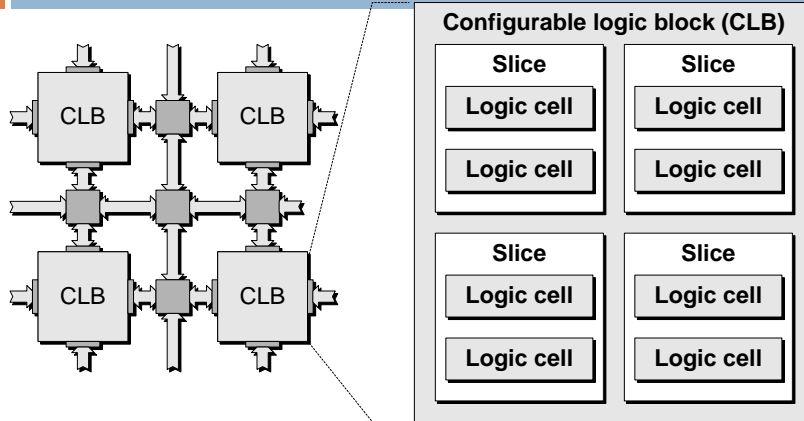
54

- Ιεραρχία στα λογικά στοιχεία
- Ένα Xilinx slice περιέχει 2 logic cells



Xilinx configurable logic block (CLB)

55



- Ένα Xilinx CLB περιέχει 4 slices και 8 logic cells
 - ο αριθμός των slices και των LCs εξαρτάται από την οικογένεια

Γρήγορες αλυσίδες κρατουμένου (fast carry chains)

56

- Τα λογικά στοιχεία περιέχουν πρόσθετη λογική για να υλοποιούν γρήγορες αλυσίδες κρατουμένου (fast carry chains)
- Μπορούμε να συνδέσουμε τις αλυσίδες 2 LCs, στη συνέχεια 2 slices, στη συνέχεια 2 CLBs ώστε να επεκτείνουμε το μέγεθός τους
- Οι γρήγορες αλυσίδες κρατουμένου βελτιώνουν την απόδοση αριθμητικών κυκλωμάτων, π.χ. αθροιστών

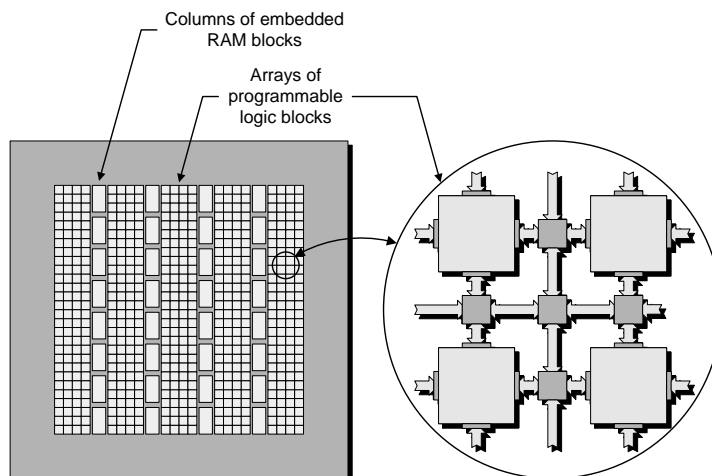
Ενσωματωμένες RAMs

57

- Λόγω της ανάγκης πολλών εφαρμογών για μεγάλη διαθέσιμη μνήμη, πολλές FPGA συσκευές ενσωματώνουν RAM μπλοκ
- Ανάλογα με την αρχιτεκτονική του FPGA τα RAM μπλοκ μπορεί να βρίσκονται στην περιφέρεια της συσκευής ή να είναι οργανωμένα σε στήλες
- Κάθε RAM μπλοκ μπορεί να χρησιμοποιηθεί ανεξάρτητα ή να συνδυαστούν πολλά μαζί ώστε να υλοποιήσουν μια μεγαλύτερη μνήμη
- Μπορούν να χρησιμοποιηθούν σε ποικιλία εφαρμογών
 - ▣ Single-port RAM
 - ▣ Dual-port RAM
 - ▣ FIFO

Ενσωματωμένες RAMs

58



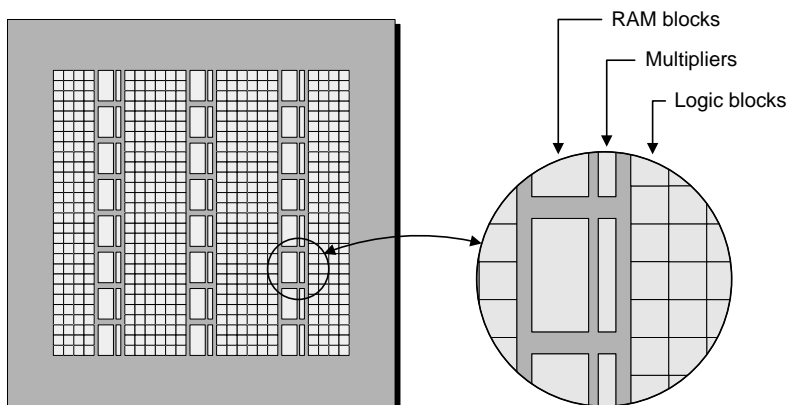
Ενσωματωμένες αριθμητικές μονάδες

59

- Κάποιες αριθμητικές συναρτήσεις, όπως ο πολλαπλασιασμός, είναι αργές όταν υλοποιηθούν από έναν αριθμό λογικών στοιχείων
- Πολλές FPGA συσκευές ενσωματώνουν τέτοιες αριθμητικές μονάδες
 - ▣ Πολλαπλασιαστές (multipliers)
 - ▣ Αθροιστές (adders)
 - ▣ Πολλαπλασιαστές-Συσσωρευτές (Multiple-Accumulate, MAC)
- Οι ενσωματωμένες αριθμητικές μονάδες μαζί με τις ενσωματωμένες μνήμες κάνουν ιδανική τη χρήση των FPGAs σε εφαρμογές ψηφιακής επεξεργασίας σήματος (digital signal processing)

Ενσωματωμένες αριθμητικές μονάδες

60



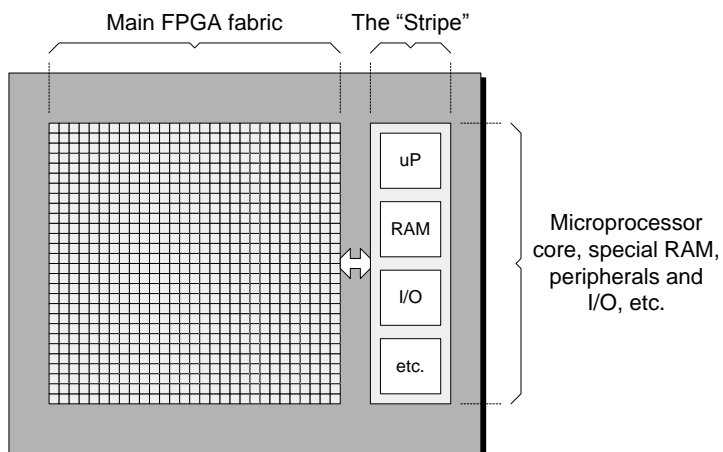
Ενσωματωμένοι επεξεργαστές

61

- Προηγμένες FPGA συσκευές ενσωματώνουν πυρήνες επεξεργαστών (microprocessor cores)
 - ▣ Χρήσιμοι σε εφαρμογές όπου απαιτείται επεξεργαστής
- Δύο τύποι πυρήνων:
 - ▣ Hard cores
 - Ενσωματώνονται στη δομή του τσιπ ως φυσικές μονάδες
 - Δεν είναι δυνατόν να τροποποιηθούν από το χρήστη
 - ▣ Soft cores
 - Υλοποιούνται από τα προγραμματιζόμενα λογικά στοιχεία του FPGA
 - Είναι δυνατόν να τροποποιηθούν και να προσαρμοστούν στις ανάγκες του χρήστη

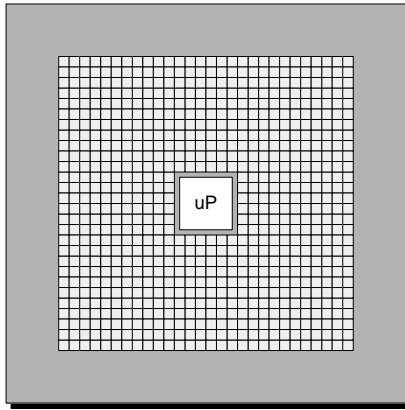
Hard microprocessor cores

62

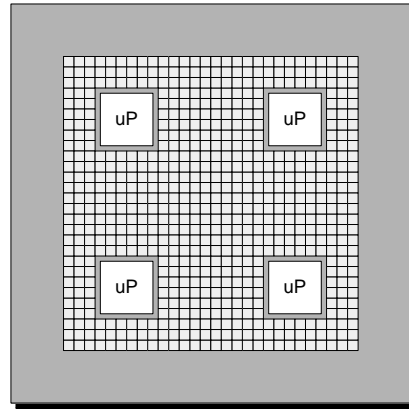


Hard microprocessor cores

63



(a) One embedded core



(b) Four embedded cores

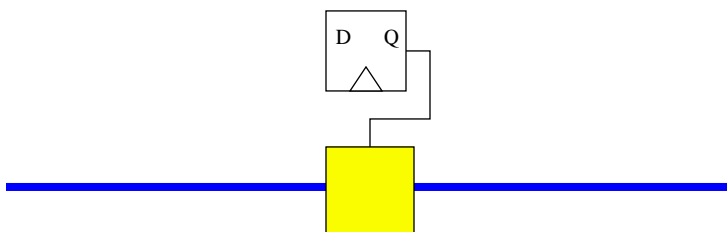
Προγραμματιζόμενη καλωδίωση

64

- Τα καλώδια οργανώνονται σε κανάλια
 - ▣ πολλά καλώδια ανά κανάλι
- Οι συνδέσεις μεταξύ των καλωδίων υλοποιούνται στα προγραμματιζόμενα σημεία διασύνδεσης
- Πρέπει να επιλεγούν:
 - ▣ κανάλια από την πηγή στον προορισμό
 - ▣ καλώδια εντός των καναλιών

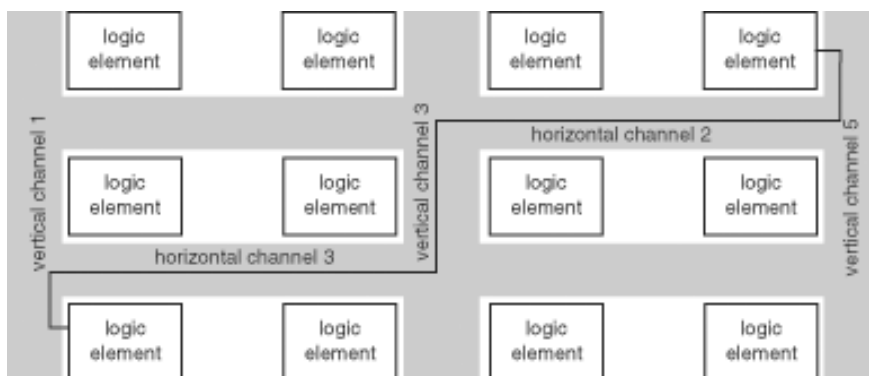
Προγραμματιζόμενο σημείο διασύνδεσης

65



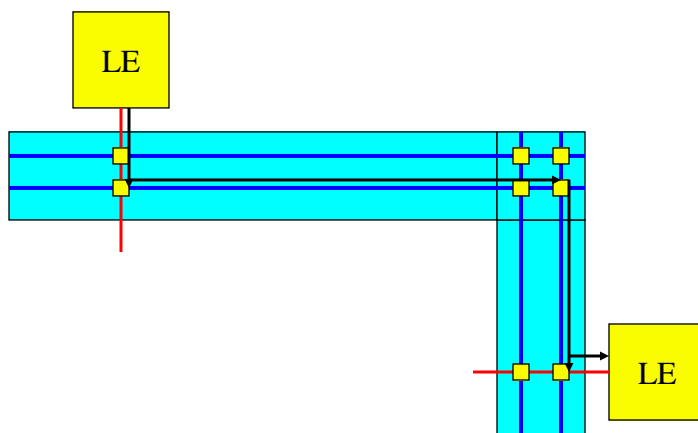
Προγραμματιζόμενα μονοπάτια καλωδίωσης

66



Επιλογή μονοπατιού

67



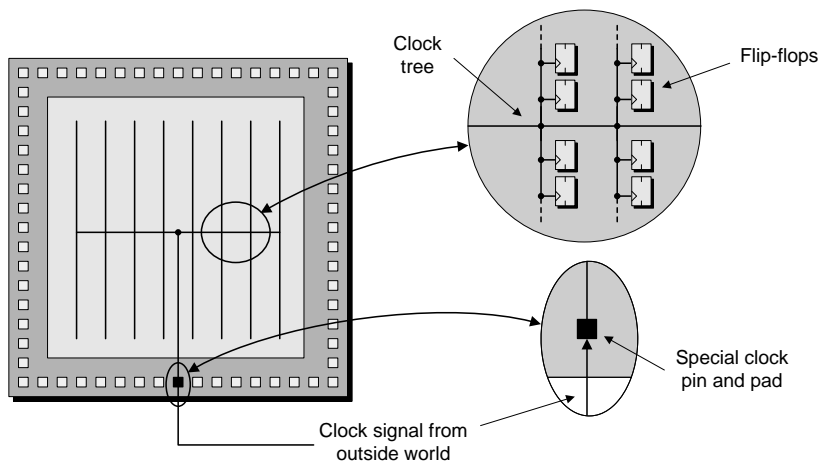
Προβλήματα δρομολόγησης

68

- Σφαιρική δρομολόγηση:
 - ▣ ποιος συνδυασμός καναλιών;
- Τοπική δρομολόγηση:
 - ▣ ποιο καλώδιο σε κάθε κανάλι;
- Μετρικό σύστημα δρομολόγησης:
 - ▣ μήκος καλωδίου
 - ▣ καθυστέρηση

Δέντρα ρολογιού (clock trees)

69



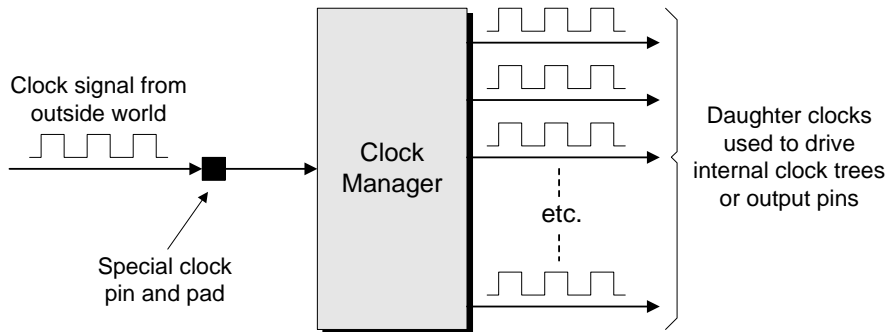
Αποφυγή απόκλισης ρολογιού (clock skew)

70

- **Clock skew:**
 - Το σήμα ρολογιού δεν φτάνει ταυτόχρονα σε όλα τα flip-flop του κυκλώματος
 - Τα flip-flop που είναι πιο κοντά στον ακροδέκτη του ρολογιού δέχονται «νωρίτερα» το ρολόι
 - Από τα **κυριότερα σχεδιαστικά προβλήματα** και στα FPGAs και στα ASICs
- **Πώς αντιμετωπίζεται το πρόβλημα στα FPGAs;**
 - Δέντρα ρολογιού (clock trees)
 - Πολλαπλοί ακροδέκτες ρολογιού (clock pins)
 - Πολλαπλά πεδία ρολογιού (clock domains)

Διευθυντές ρολογιού (clock managers)

71



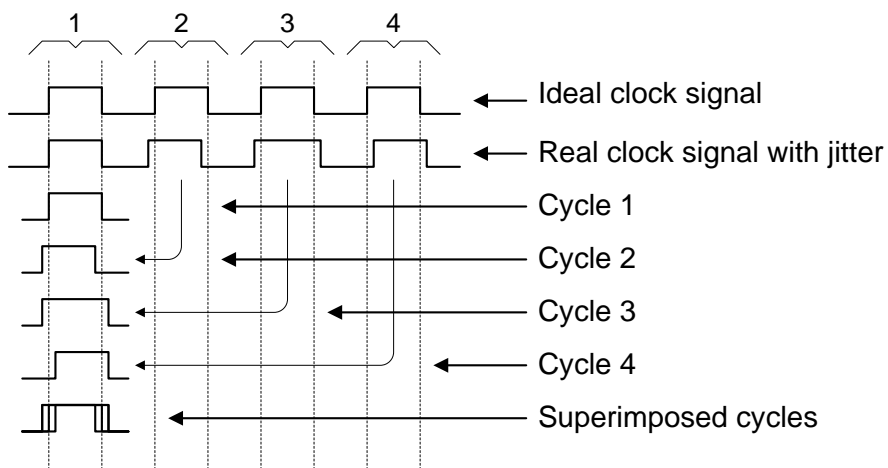
Clock managers

72

- Παράγουν ρολόγια που τροφοδοτούν είτε την εσωτερική λογική (δέντρα ρολογιού) είτε την εξωτερική λογική
- Παρέχουν κάποιες από τις ακόλουθες λειτουργίες
 - ▣ Αφαίρεση παραμόρφωσης χρονισμού (jitter removal)
 - ▣ Σύνθεση συχνότητας (frequency synthesis)
 - ▣ Ολίσθηση φάσης (phase shifting)
 - ▣ Διόρθωση αυτό-απόκλισης (auto-skew correction)

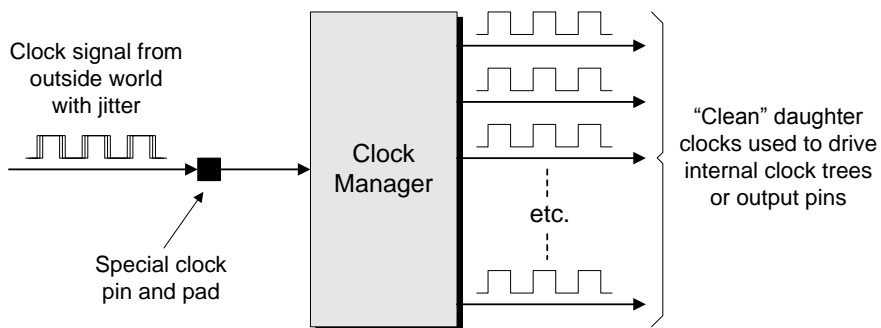
Φαινόμενο παραμόρφωσης (jitter)

73



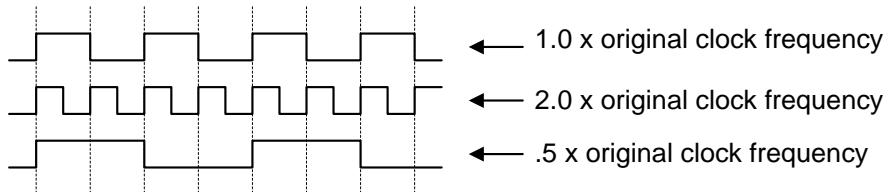
Jitter removal

74



Frequency synthesis

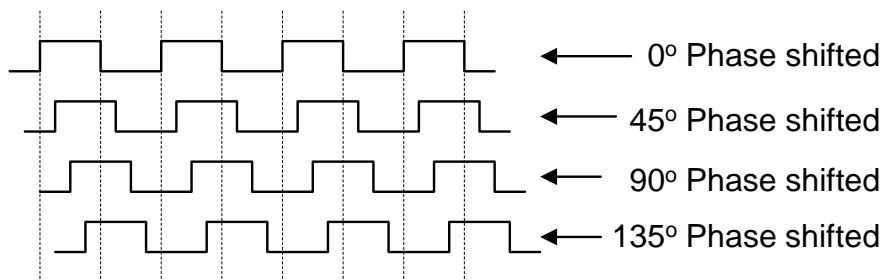
75



- Ένας clock manager μπορεί να παράγει:
 - Πολλαπλάσια και υπο-πολλαπλάσια της αρχικής συχνότητας
 - Δεκαδικά πολλαπλάσια της αρχικής συχνότητας, π.χ. $3/5 \times f$

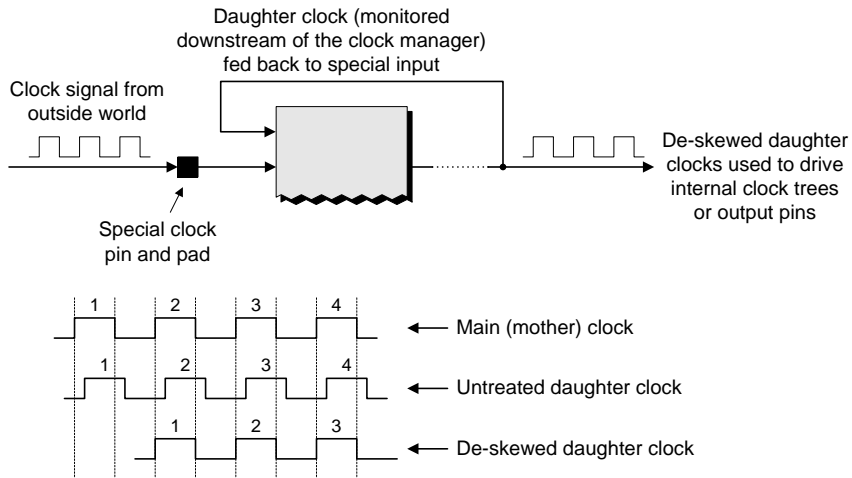
Phase shifting

76



Auto-skew correction

77



Ακροδέκτες εισόδου/εξόδου (I/O pins)

78

- Βασική επιλογή:
είσοδος, έξοδος, τρισταθής λογική
- Επιπλέον προγραμματιζόμενα χαρακτηριστικά:
 - καταχωρητής (register)
 - πρότυπα λογικής (logic standard)
 - ταχύτητα μετάβασης (edge rate)
 - αντίσταση τερματισμού (terminating resistor)

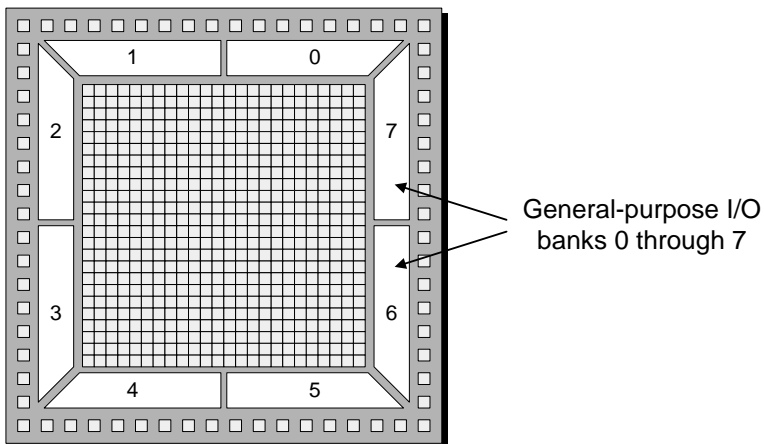
Πρότυπα εισόδου/εξόδου (I/O standards)

79

- Ο σχεδιαστής πρέπει να επιλέξει ένα συγκεκριμένο πρότυπο I/O
 - ▣ Ηλεκτρικά χαρακτηριστικά των σημάτων, π.χ. επίπεδα λογικής 0 και 1
 - ▣ Ανάλογα με την εφαρμογή, τις συσκευές που επικοινωνεί το FPGA, κτλ.
- Ο σχεδιαστής μπορεί να χρειαστεί να επιλέξει διαφορετικό πρότυπο για διαφορετικά I/O

Πολλαπλές συστοιχίες I/O (I/O banks)

80



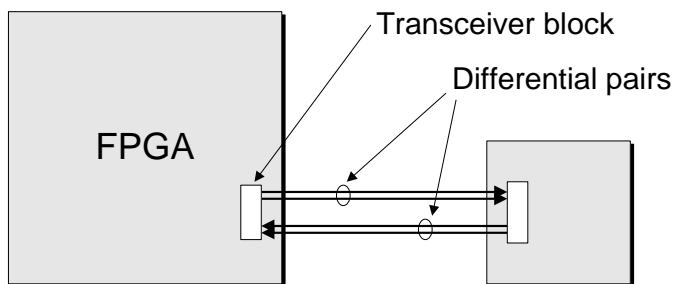
Τροφοδοσία πυρήνα εναντίον τροφοδοσία I/O

81

- Πολλαπλοί ακροδέκτες χρησιμοποιούνται για την τροφοδοσία του FPGA
 - ▣ Για την παροχή πολλαπλών τάσεων τροφοδοσίας (supply voltages)
 - Για την τροφοδοσία της λογικής του FPGA (core voltage)
 - Για την υποστήριξη των διαφορετικών προτύπων I/O (I/O supply voltages)

Πομποδέκτες υψηλής ταχύτητας (gigabit transceivers)

82



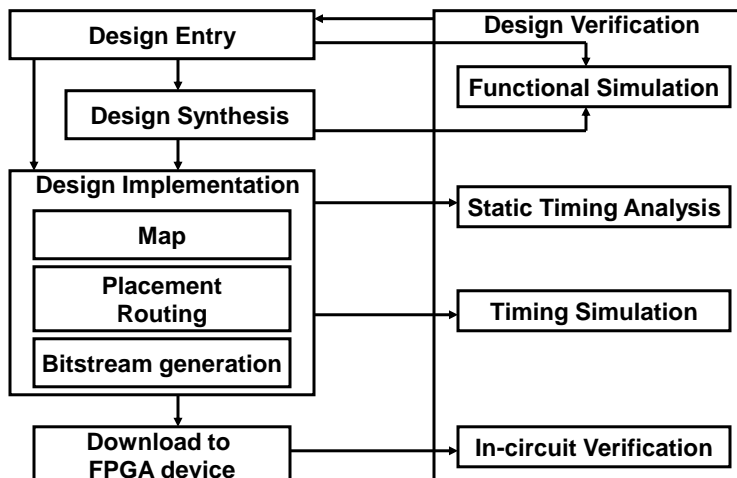
Ενότητα 5

83

- Ροή σχεδίασης με τη χρήση FPGAs
- Ποια είναι τα βήματα της σχεδίασης ενός ψηφιακού κυκλώματος με τη χρήση ενός FPGA ;

Ροής σχεδίασης με τη χρήση FPGAs

84



Σύνθεση (synthesis)

85

- Σύνθεση & βελτιστοποίηση της σχεδίασης:
 - ▣ η περιγραφή συμπεριφοράς (behavioral description) μεταφράζεται σε περιγραφή δομής (structural netlist)
- Εργαλείο σύνθεσης:
 - ▣ από τον κατασκευαστή του FPGA
 - ▣ από ανεξάρτητο κατασκευαστή λογισμικού
- Μεταφραστές (netlist translators)
 - ▣ μεταφράζουν την περιγραφή δομής σε μορφή που υποστηρίζει το εργαλείο του κατασκευαστή

Υλοποίηση (implementation)

86

- **Αντιστοίχιση (map)**
 - ▣ Πώς τεμαχίζουμε τη συνάρτηση σε λογικά στοιχεία;
 - ▣ Πώς υλοποιούμε μία λειτουργία στο εσωτερικό ενός λογικού στοιχείου;
- **Τοποθέτηση (placement)**
 - ▣ Πού τοποθετούμε κάθε κομμάτι λογικής στη διάταξη των λογικών στοιχείων;
- **Δρομολόγηση (routing)**
 - ▣ Ποιες καλωδιώσεις χρησιμοποιούνται για να συνδεθούν τα λογικά στοιχεία;

Περιορισμοί (constraints)

87

- Μπορούμε να εισάγουμε περιορισμούς:
 - στην αντιστοίχιση (mapping constraints)
 - να καθορίσουμε τον τρόπο που ένα κομμάτι λογικής θα αντιστοιχηθεί σε ένα λογικό στοιχείο
 - στην τοποθέτηση (placement constraints)
 - να καθορίσουμε την ακριβή θέση (ή τις πιθανές θέσεις) ενός λογικού στοιχείου
 - χρονικούς περιορισμούς (timing constraints)
 - χρονικές προδιαγραφές για τα μονοπάτια της σχεδίασης (π.χ. περίοδο ρολογιού, καθυστέρηση από είσοδο σε καταχωρητή, καθυστέρηση από καταχωρητή σε έξοδο)

Πιστοποίηση (verification)

88

- Η συνάρτηση και η απόδοση της σχεδίασης πιστοποιείται με τρεις τρόπους:
 - προσομοίωση (simulation)
 - προσομοίωση συνάρτησης
 - χρονική προσομοίωση
 - χρονική ανάλυση (timing analysis)
 - στο κύκλωμα (in-circuit verification)

Προσομοίωση συνάρτησης (functional simulation)

89

- Πιστοποιεί την ορθότητα της σχεδίασης πριν υλοποιηθεί
- Εκτελείται στα πρώτα στάδια της σχεδίασης
 - ▣ είτε σε περιγραφή συμπεριφοράς είτε σε περιγραφή δομής
- Χρησιμοποιεί μοναδιαία καθυστέρηση (unit delay)
 - ▣ χρονική πληροφορία δεν είναι διαθέσιμη

Χρονική προσομοίωση (timing simulation)

90

- Πιστοποιεί ότι η σχεδίαση λειτουργεί στην επιθυμητή ταχύτητα
- Εκτελείται μετά την υλοποίηση
- **Απαραίτητη**
 - ▣ καθορίζει τα κρίσιμα μονοπάτια κάτω από τις χειρότερες συνθήκες
 - ▣ ανιχνεύει χρονικές παραβιάσεις (setup, hold violations)

Χρονική ανάλυση (timing analysis)

91

- Υπολογίζει τις καθυστερήσεις των μονοπατιών του κυκλώματος
- Εκτελείται μετά την υλοποίηση
- Χρήσιμη γιατί:
 - ▣ πιστοποιεί ότι η σχεδίαση ικανοποιεί τις χρονικές προδιαγραφές
 - ▣ πληροφορεί το σχεδιαστή για τα «αργά» μονοπάτια
 - ▣ προσφέρει υλικό για την τεκμηρίωση της σχεδίασης

Πιστοποίηση στο κύκλωμα (in-circuit verification)

92

- Πιστοποιεί ότι η σχεδίαση λειτουργεί στην τελική εφαρμογή (κάτω από πραγματικές συνθήκες λειτουργίας)
- Οι κατασκευαστές παρέχουν εργαλεία για να βοηθήσουν την πιστοποίηση στο κύκλωμα

Διαμόρφωση (configuration)

93

- Πρέπει να θέσει τα ψηφία ελέγχου για:
 - ▣ λογικά στοιχεία
 - ▣ σημεία διασύνδεσης
 - ▣ ακροδέκτες εισόδου/εξόδου
- Συνήθως διαμορφώνεται εκτός-λειτουργίας (off-line configuration)
 - ▣ ξεχωριστό στάδιο καψίματος (burn-in) – antifuse
 - ▣ κατά την εκκίνηση -- SRAM

Επαναδιαμόρφωση (reconfiguration)

94

- Μερικές συσκευές FPGAs επιτρέπουν γρήγορη διαμόρφωση
 - ▣ μερικούς κύκλους ρολογιού, όχι χιλιάδες κύκλους ρολογιού
- Επιτρέπουν το υλικό να αλλάξει on-the-fly

Διαμόρφωση SRAM-based FPGAs

- = I/O pin/pad
- ▭ = SRAM cell

- Τα SRAM cells μπορούν να αντιμετωπιστούν σαν ένας μακρύς καταχωρητής ολίσθησης
- Ισχύει μόνο για σειριακούς μηχανισμούς διαμόρφωσης

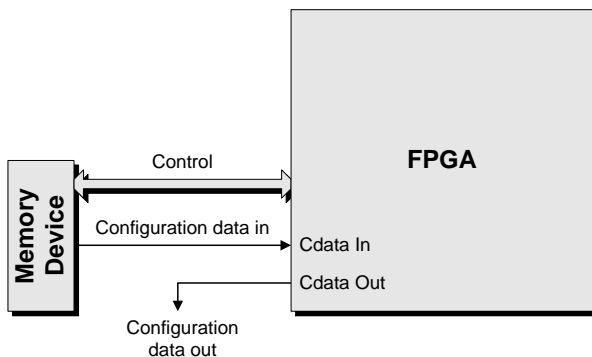
Θύρα διαμόρφωσης (configuration port)

- **Configuration mode pins:**
 - Καθορίζουν ποιος τύπος διαμόρφωσης θα χρησιμοποιηθεί
- **Configuration data in:**
 - Τροφοδοτεί την ακολουθία διαμόρφωσης (configuration bitstream)
- **Configuration data out:**
 - Διαβάζει πίσω την ακολουθία διαμόρφωσης
- **Configuration control pins:**
 - Ελέγχουν τη φόρτωση της ακολουθίας διαμόρφωσης

Mode Pins	Mode
0 0	Serial load with FPGA as master
0 1	Serial load with FPGA as slave
1 0	Parallel load with FPGA as master
1 1	Parallel load with FPGA as slave

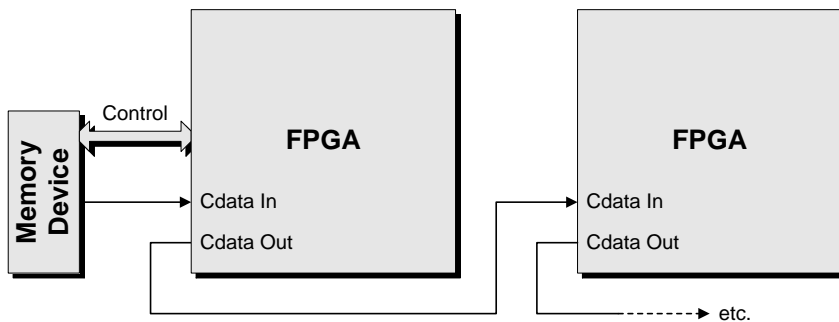
Serial load – FPGA master

97



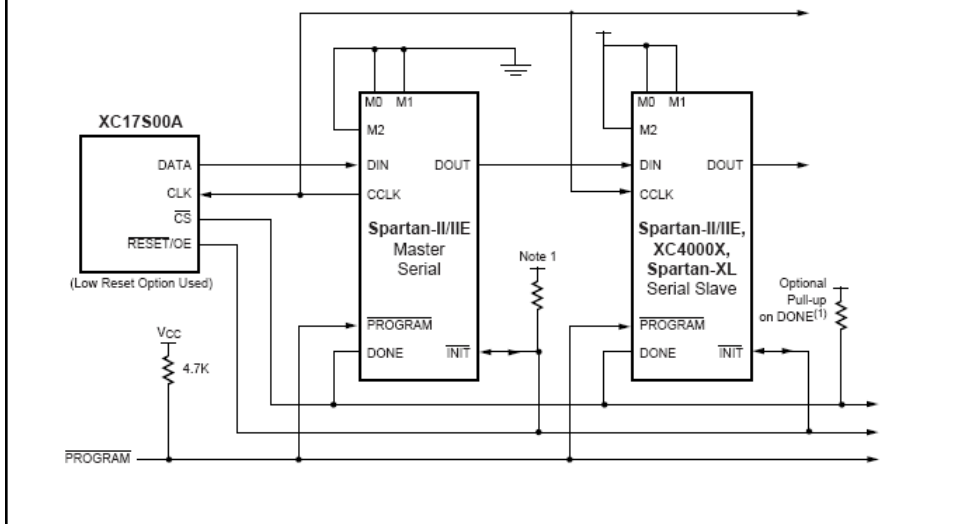
Αλυσίδα διαμόρφωσης

98



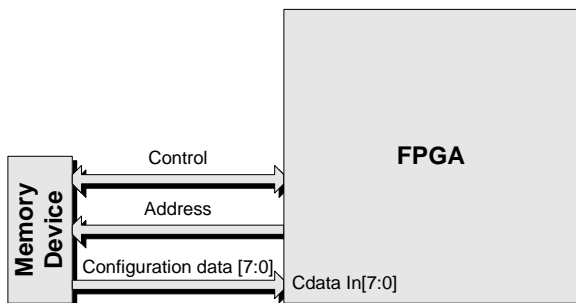
Παράδειγμα: Spartan II, σειριακή διαμόρφωση

99



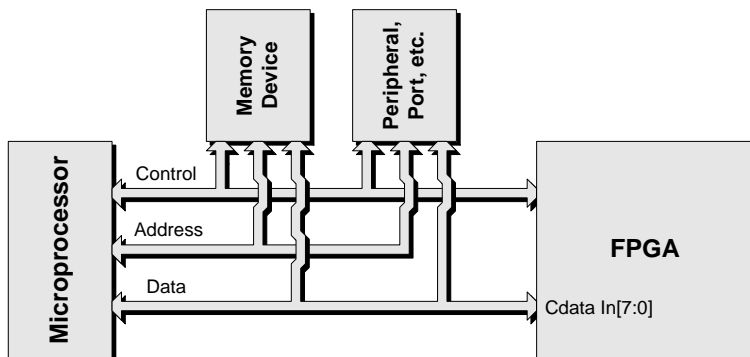
Parallel load – FPGA master

100



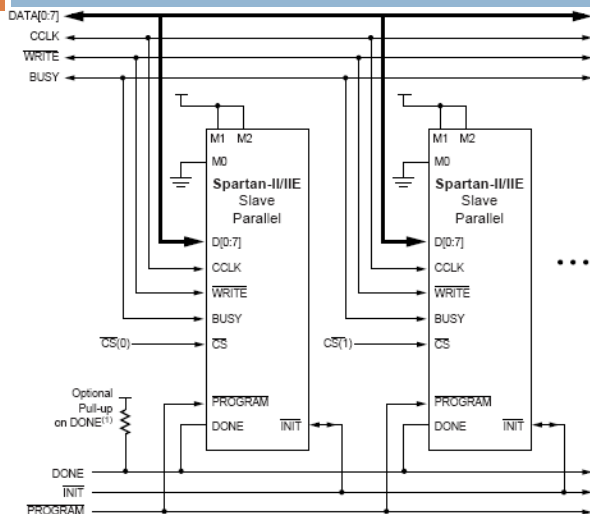
Parallel load – FPGA slave

101



Παράδειγμα: Spartan II, παράλληλη διαμόρφωση

102



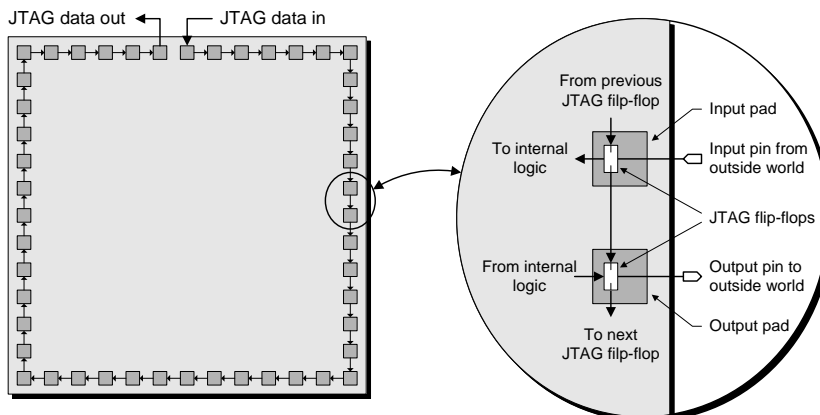
JTAG port

103

- JTAG: Joint Test Action Group
- Αρχικά σχεδιάστηκε για την υλοποίηση της σάρωσης ορίων (boundary scan):
 - ▣ παρέχει αλυσίδα σάρωσης (scan chain) στους ακροδέκτες
- Στα σημερινά FPGAs χρησιμοποιείται και για διαμόρφωση
 - ▣ συνδέει τον καταχωρητή ολίσθησης των SRAM cells με την αλυσίδα σάρωσης

JTAG scan chain

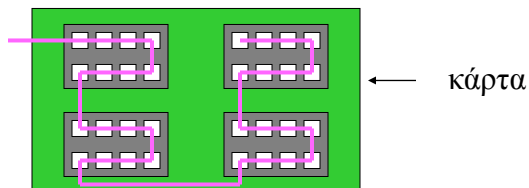
104



Δοκιμή του ολοκληρωμένου στην κάρτα με χρήση του JTAG

105

- Αποσυνδέει το ολοκληρωμένο από την υπόλοιπη κάρτα για δοκιμή
- Χρησιμοποιείται για τη δοκιμή (testing) της κάρτας και των ολοκληρωμένων κυκλωμάτων



106

Ενότητα 6

107

- Παρουσίαση μιας συγκεκριμένης αρχιτεκτονικής FPGAs
- SRAM-based FPGAs
 - ▣ Xilinx Spartan-3

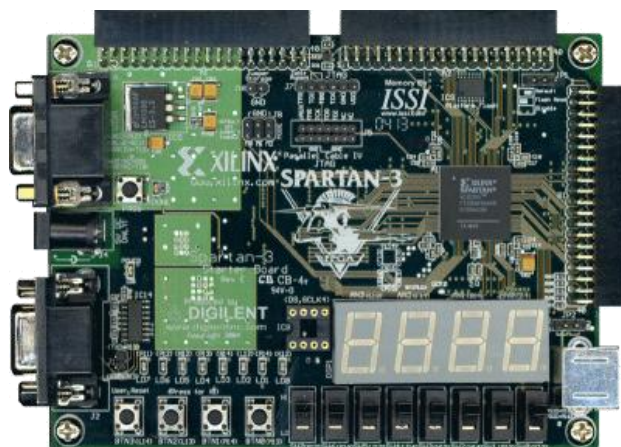
Οικογένειες Xilinx Spartan

108

- Η εταιρεία Xilinx με τη σειρά Spartan παρέχει χαμηλού κόστους FPGA συσκευές
- Οι περισσότερες συσκευές Spartan σχετίζονται στενά με κάποιο άλλο προϊόν της Xilinx
- Υπάρχουν πολλές οικογένειες Spartan FPGAs:
 - ▣ Spartan-II, Spartan-IIE (παρόμοια με Virtex)
 - ▣ **Spartan-3**, Spartan-3E (παρόμοια με Virtex-4)

Αναπτυξιακή πλατφόρμα: Digilent Spartan-3

109



Πίνακας των προϊόντων Spartan-3

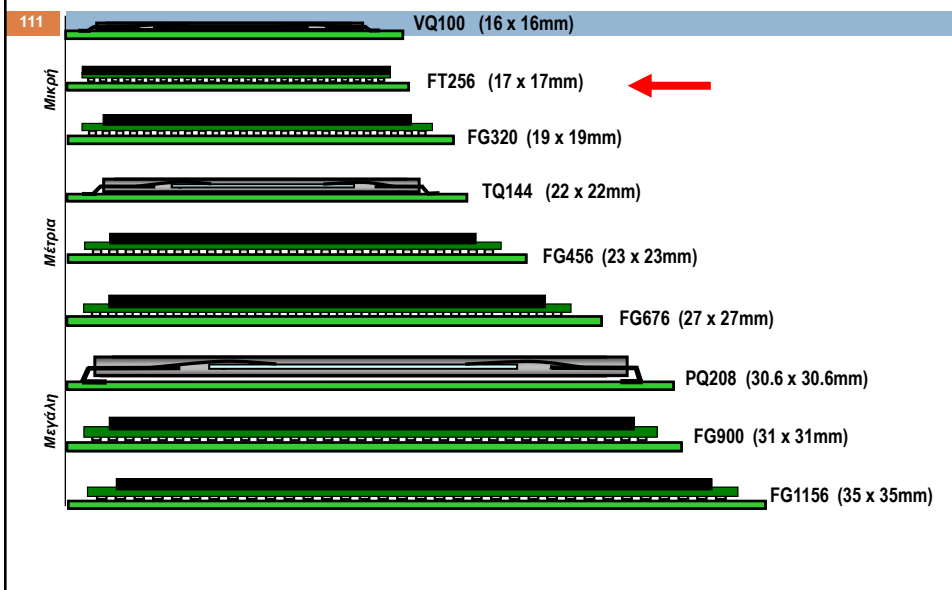
110

← 100X διάστημα πυκνότητας →

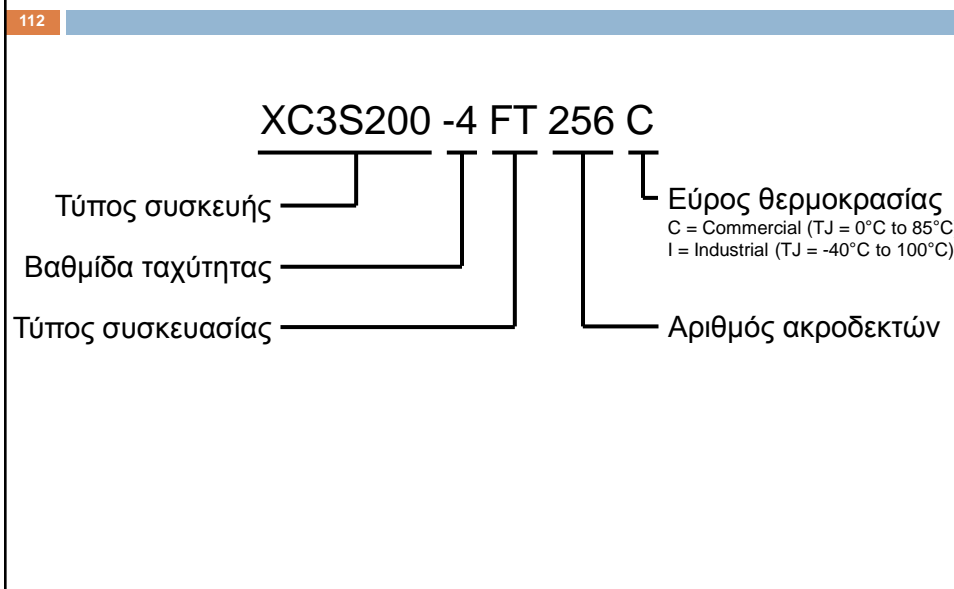
Device	XC3S50	XC3S200	XC3S400	XC3S1000	XC3S1500	XC3S2000	XC3S4000	XC3S5000
System Gates	50K	200K	400K	1000K	1500K	2000K	4000K	5000K
Logic Cells	1,728	4,320	8,064	17,280	29,952	46,080	62,208	74,880
Dedicated Multipliers	4	12	16	24	32	40	96	104
Block RAM Blocks	4	12	16	24	32	40	96	104
Block RAM Bits	72K	216K	288K	432K	576K	720K	1,728K	1,872K
Distributed RAM Bits	12K	30K	56K	120K	208K	320K	432K	520K
DCMs	2	4	4	4	4	4	4	4
I/O Standards	24	24	24	24	24	24	24	24
Max Single Ended I/O	124	173	264	391	487	565	712	784

50.000 έως 5.000.000 System Gates

Επιλογή συσκευασίας



Κωδικός παραγγελίας



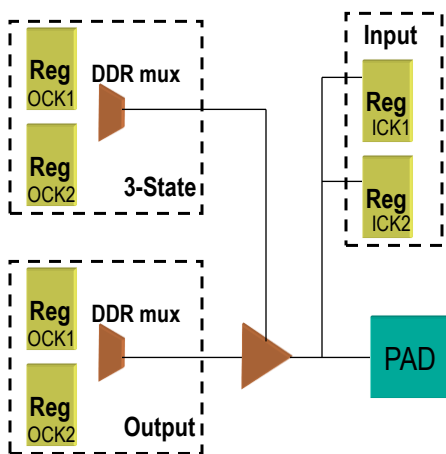
Οικογένεια Xilinx Spartan-3

113

- Προγραμματιζόμενα μπλοκ εισόδου/εξόδου
 - ▣ Input Output Blocks (IOB)
- Διευθυντές ρολογιού
 - ▣ Digital Clock Manager (DCM)
- Προγραμματιζόμενα μπλοκ λογικής
 - ▣ Configurable Logic Blocks (CLB)
- Ευέλικτα μπλοκ μνήμης
 - ▣ BlockRAM
- Προγραμματιζόμενες πηγές δρομολόγησης

I/O Block (IOB)

114

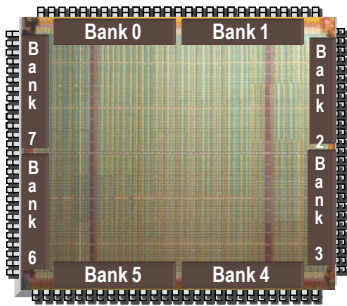


- Μονοπάτι εισόδου
 - 2 DDR registers
- Μονοπάτι εξόδου
 - 2 DDR registers
 - 2 3-state DDR registers
- Ξεχωριστά ρολόγια

DDR = Double Data Rate

Δυνατότητες σύνδεσης

115



- Μονοί και διαφορικοί ακροδέκτες
 - 784 μονοί, 344 διαφορεικά ζεύγη
 - 622 Mb/sec LVDS
 - 24 I/O πρότυπα, 8 I/O banks
 - Υποστήριξη PCI 32/33 and 64/33
 - Περιορίζει την ανάγκη για ακριβούς πομποδέκτες (bus transceivers)
- 3.3V, 2.5V, 1.8V, 1.5V, 1.2V

Διασύνδεση Chip-με-Chip:

LVDS LVC MOS LV TTL

Διασύνδεση με πλακέτα βάσης (backplane):

GTL GTL+ PCI BLVDS

Διασύνδεση με μνήμη υψηλής ταχύτητας:

HSTL SSTL

Επιλογή πρότυπου I/O

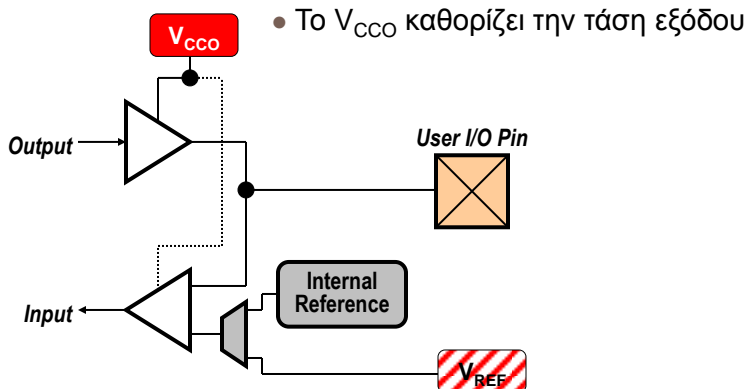
116

	Standard	Output V_{CCO}	Input V_{REF}
Single ended	LV TTL	3.3V	--
	LVC MOS33	3.3V	--
	LVC MOS25	2.5V	--
	LVC MOS18	1.8V	--
	LVC MOS15	1.5V	--
	LVC MOS12	1.2V	--
	PCI 32/64 bit 33MHz	3.3V	--
	SSTL2 Class I	2.5V	1.25V
	SSTL2 Class II	2.5V	1.25V
	SSTL18 Class I	1.8V	0.9V
	HSTL Class I	1.5V	0.75V
	HSTL Class III	1.5V	0.9V
	HSTL18 Class I	1.8V	0.9V
	HSTL18 Class II	1.8V	0.9V
	HSTL18 Class III	1.8V	1.1V
	Differential	GTL	--
GTL+		--	1.0V
LVDS2.5		2.5V	--
Bus LVDS2.5		2.5V	--
Ultra LVDS2.5		2.5V	--
LVDS_ext2.5		2.5V	--
RSDS		2.5V	--
LDT2.5		2.5V	--

- Περισσότερα πρότυπα επιτρέπουν περισσότερες δυνατότητες ολοκλήρωσης στο σύστημα

Επιλογή πρότυπου I/O

117

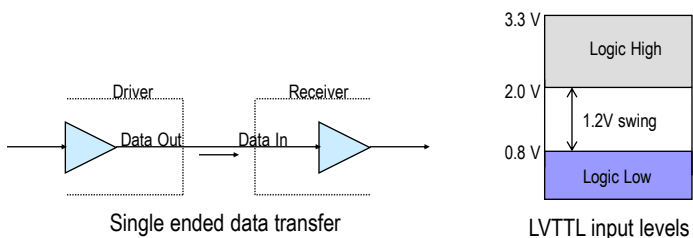


- Το V_{REF} καθορίζει την τάση αναφοράς για το κατώφλι εισόδου
- Διαθέσιμο σαν I/O του χρήστη όταν χρησιμοποιείται εσωτερική τάση αναφοράς

Μονοί ακροδέκτες

118

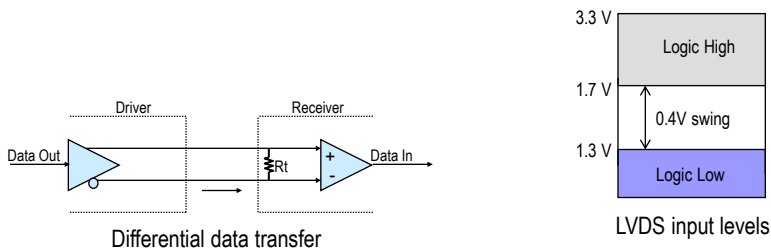
- Τα δεδομένα μεταφέρονται σε μια μονή γραμμή
- Μεγάλη αιώρηση τάσης μεταξύ των λογικών επιπέδων



Διαφορικοί ακροδέκτες

119

- Ένα bit δεδομένων μεταφέρεται σε δυο γραμμές
- Η διαφορά τάσης καθορίζει το λογικό επίπεδο
- Μικρή αιώρηση τάσης μεταξύ των λογικών επιπέδων



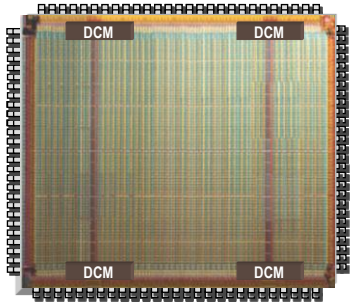
Οικογένεια Xilinx Spartan-3

120

- Προγραμματιζόμενα μπλοκ εισόδου/εξόδου
 - ▣ Input Output Blocks (IOB)
- Διευθυντές ρολογιού
 - ▣ Digital Clock Manager (DCM)
- Προγραμματιζόμενα μπλοκ λογικής
 - ▣ Configurable Logic Blocks (CLB)
- Ευέλικτα μπλοκ μνήμης
 - ▣ BlockRAM
- Προγραμματιζόμενες πηγές δρομολόγησης

Digital Clock Manager (DCM)

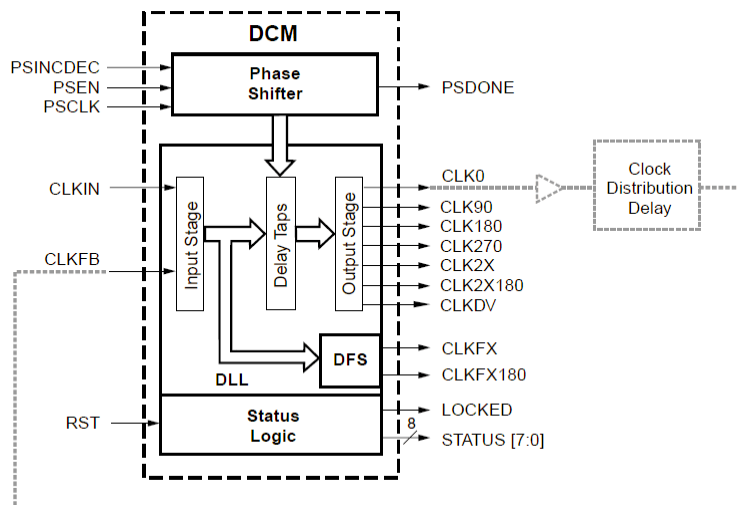
121



- 4 DCMs
- Βασικές λειτουργίες που παρέχει κάθε DCM:
 - Περιορισμός της απόκλισης του ρολογιού (clock skew)
 - Ολίσθηση φάσης – Digital Phase Shift (DPS)
 - Σύνθεση συχνοτήτων – Digital Frequency Synthesis (DFS)

Διάγραμμα ενός DCM

122



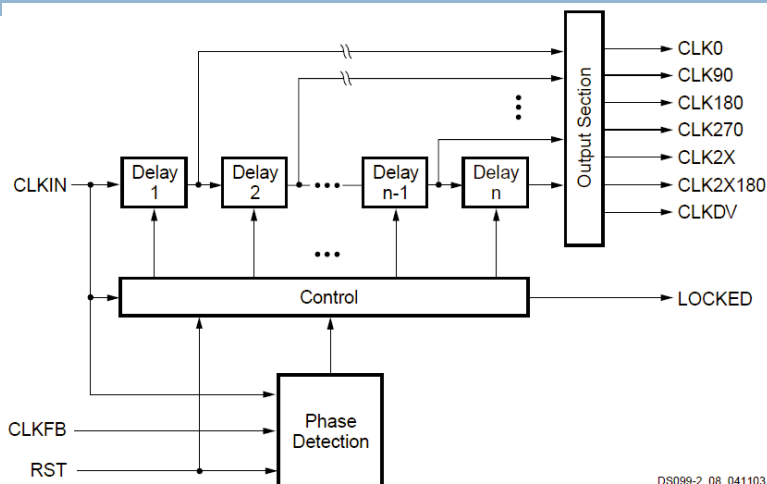
Delay Locked Loop (DLL) – Βρόχος σταθερής καθυστέρησης

123

- Εισάγει καθυστέρηση στο δίκτυο του ρολογιού έως ότου η ανοδική ακμή του ρολογιού εισόδου (CLKIN) είναι σε φάση με την ανοδική ακμή του ρολογιού ανάδρασης
- Με ένα καλοσχεδιασμένο δίκτυο διανομής του ρολογιού, οι ακμές του ρολογιού «φτάνουν» ταυτόχρονα σε όλα τα σημεία της συσκευής, μαζί με την άφιξη της ακμή του ρολογιού εισόδου του DLL

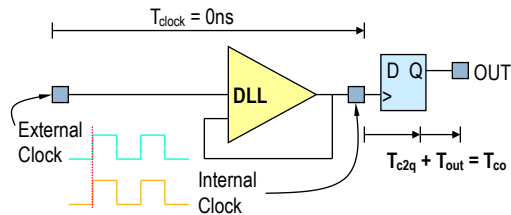
Διάγραμμα του DLL

124



DLL: προσαρμογή χρονισμού ρολογιών

125

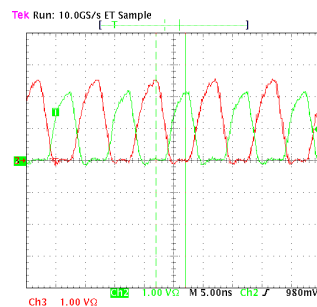
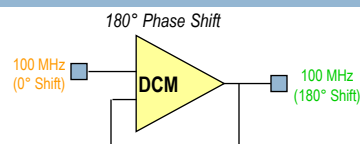


- Περιορίζει την καθυστέρηση της διανομής του ρολογιού
 - ▣ Το εξωτερικό ρολόι και το εσωτερικό ρολόι είναι «ευθυγραμμισμένα»
- Προαιρετική διόρθωση του κύκλου λειτουργίας (duty cycle)
 - ▣ 50/50 κύκλος λειτουργίας

DLL: ολίσθηση φάσης

126

- Επιλογές ολίσθησης φάσης: 0° , 90° , 180° , και 270° .
- Χρήση ρολογιών με διαφορετικές φάσεις: τεχνική για αύξηση της απόδοσης
- Κατάλληλο για διασύνδεση εξωτερικών μνημών



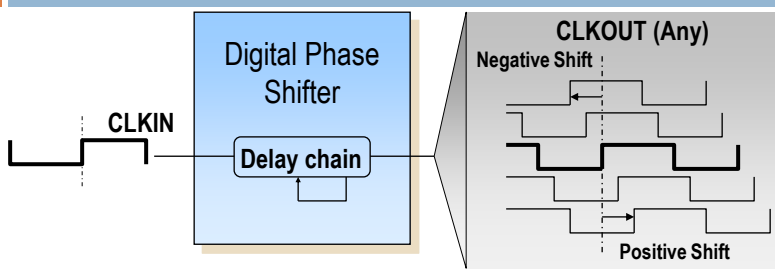
DLL: προσαρμογή συχνότητας

127

- Πολλαπλασιασμός συχνότητας x2
- Διαίρεση συχνότητας
 - ▣ Επιλογή τιμής διαίρεσης: από 1.5 έως 16
- Επιλογή διόρθωσης του κύκλου λειτουργίας σε 50/50

Digital Phase Shifter (DPS)

128



- Τοποθετεί την ακμή του ρολογιού οπουδήποτε εντός της περιόδου
 - ▣ Ολίσθηση φάσης = $(PS/256) \times \text{period}$, $-255 < PS < +255$
- Επηρεάζει τις εξόδους όλων των DCMs

Digital Frequency Synthesizer (DFS)

129

CLKIN

Digital Frequency Synthesizer

Period Calculator

DFS Outputs

CLKIN (for Reference)

CLKFX

CLKFX180

e.g. M = 3, D = 1

- Συνθέτει οποιαδήποτε συχνότητα εντός του εύρους λειτουργίας:
 - ▣ $CLKOUT = (M \div D) \times CLKIN$
 - ▣ M = 2 to 32 and D = 1 to 32
- Οι έξοδοι έχουν κύκλο λειτουργίας 50/50

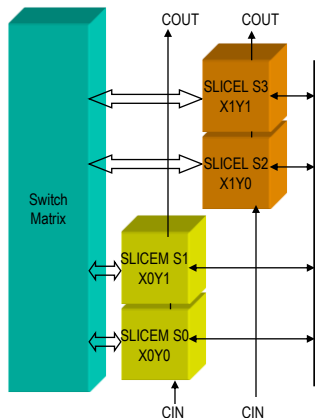
Οικογένεια Xilinx Spartan-3

130

- Προγραμματιζόμενα μπλοκ εισόδου/εξόδου
 - ▣ Input Output Blocks (IOB)
- Διευθυντές ρολογιού
 - ▣ Digital Clock Manager (DCM)
- Προγραμματιζόμενα μπλοκ λογικής
 - ▣ Configurable Logic Blocks (CLB)
- Ευέλικτα μπλοκ μνήμης
 - ▣ BlockRAM
- Προγραμματιζόμενες πηγές δρομολόγησης

Configurable Logic Block (CLB)

131



- Η μήτρα μεταγωγής (switch matrix) συνδέει στις πηγές δρομολόγησης
- 4 slices ανά CLB.
 - 2 SLICEL είναι μόνον λογική
 - 2 SLICEM είναι λογική/μνήμη
- Αριθμητικές συναρτήσεις με αλυσίδες πρόβλεψης κρατουμένου

Βασική δομή ενός slice του Spartan-3

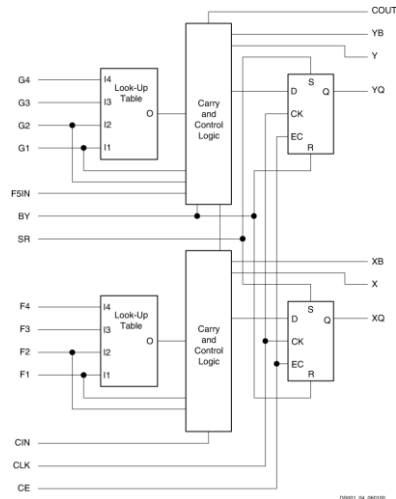
132

- SLICEL : περιέχει δύο 4-εισόδων LUTs ακολουθούμενα από D flip-flop
- SLICEM : όπως του SLICEL, αλλά τα LUTs μπορούν να χρησιμοποιηθούν ως RAM ή ως καταχωρητής ολίσθησης (shift register)

Βασική δομή ενός slice του Spartan-3

133

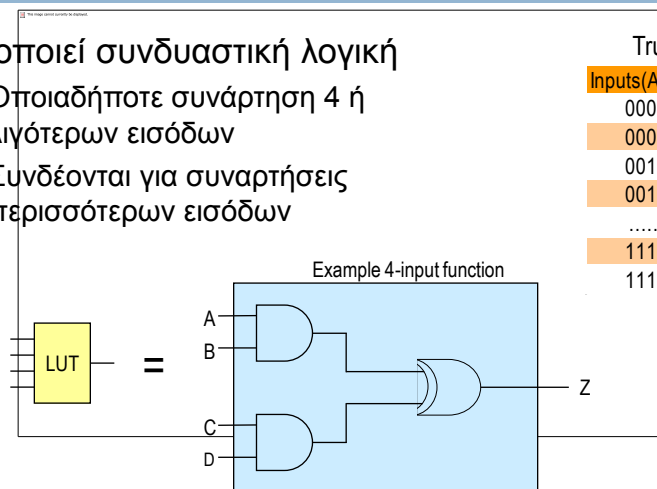
- 4-εισόδων LUT
 - Οποιαδήποτε λογική συνάρτηση 4-εισόδων
 - 16-bit x 1 RAM (SLICEM)
 - 16-bit shift register (SLICEM)
- Κρατούμενο & λογική ελέγχου
 - Γρήγορη αριθμητική λογική
 - Λογική πολλαπλασιαστή
 - Λογική πολυπλέκτη
- Στοιχείο αποθήκευσης
 - Latch ή flip-flop
 - Set και reset
 - Κανονικές ή αντεστραμμένες εισόδους



4-εισόδων LUT

134

- Υλοποιεί συνδυαστική λογική
 - Οποιαδήποτε συνάρτηση 4 ή λιγότερων εισόδων
 - Συνδέονται για συναρτήσεις περισσότερων εισόδων

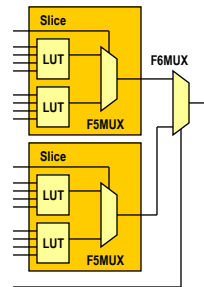


Truth Table	
Inputs(ABCD)	Output(Z)
0000	0
0001	0
0010	1
0011	0
...	..
1110	1
1111	1

Πολυπλέκτες

135

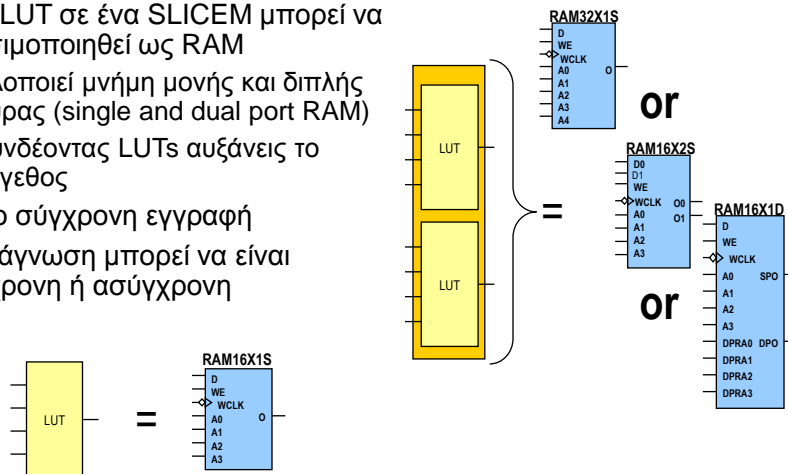
- Πιο αποδοτική από την υλοποίηση με LUTs
 - ▣ F5MUX με τις εξόδους του LUT
 - ▣ F6MUX με τις εξόδους του SLICE
 - ▣ F7MUX με τις εξόδους του CLB
 - ▣ F8MUX με τις εξόδους του F7MUX
- Αποδοτικός τρόπος για να υλοποιήσεις μεγάλους πολυπλέκτες και συναρτήσεις μέχρι 8 εισόδους



Κατανεμημένη RAM

136

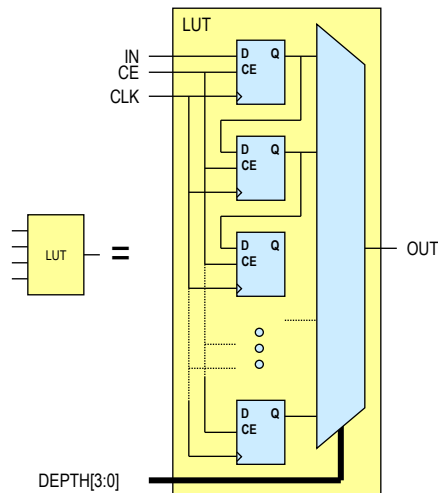
- Ένα LUT σε ένα SLICEM μπορεί να χρησιμοποιηθεί ως RAM
 - ▣ Υλοποιεί μνήμη μονής και διπλής θύρας (single and dual port RAM)
 - ▣ Συνδέοντας LUTs αυξάνεις το μέγεθος
- Μόνο σύγχρονη εγγραφή
- Η ανάγνωση μπορεί να είναι σύγχρονη ή ασύγχρονη



Καταχωρητής ολίσθησης

137

- Ένα LUT σε ένα SLICEM μπορεί να χρησιμοποιηθεί ως καταχωρητής ολίσθησης
 - ▣ Συνδέοντας LUTs αυξάνεις το μέγεθος



Οικογένεια Xilinx Spartan-3

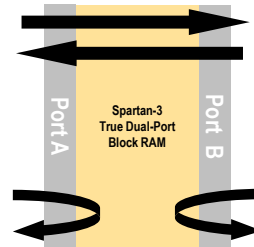
138

- Προγραμματιζόμενα μπλοκ εισόδου/εξόδου
 - ▣ Input Output Blocks (IOB)
- Διευθυντές ρολογιού
 - ▣ Digital Clock Manager (DCM)
- Προγραμματιζόμενα μπλοκ λογικής
 - ▣ Configurable Logic Blocks (CLB)
- Ευέλικτα μπλοκ μνήμης
 - ▣ BlockRAM
- Προγραμματιζόμενες πηγές δρομολόγησης

BlockRAM

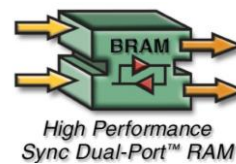
139

- Μπλοκ σύγχρονης RAM των 18-kilobit
- Ιδανικά για υλοποιήσεις διαφορετικών τύπων μνημών
- Υλοποιούν μνήμες μονής (single port) και διπλής θύρας (dual port)
 - ▣ Ιδανικές για υλοποίηση FIFOs
- Μπορούν να αρχικοποιηθούν και να χρησιμοποιηθούν ως σύγχρονες ROM



BlockRAM

- Ανεξάρτητη διευθέτηση των θυρών A και B
- Επιτρέπει μετατροπή του εύρους των δεδομένων

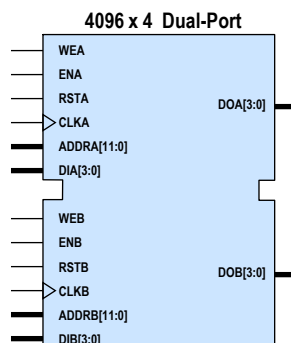


Configuration	Depth	Data bits	Parity bits
16K x 1	16Kb	1	0
8K x 2	8Kb	2	0
4K x 4	4Kb	4	0
2K x 9	2Kb	8	1
1K x 18	1Kb	16	2
512 x 36	512	32	4

Dual-port RAM

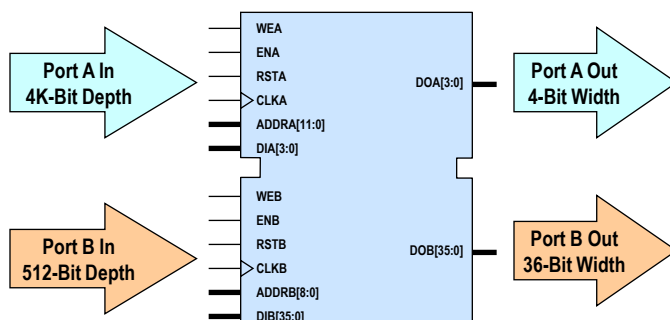
141

- Επιτρέπει ταυτόχρονη ανάγνωση και εγγραφή και στις δυο θύρες
- Κάθε θύρα έχει ανεξάρτητα σήματα ελέγχου:
 - Address
 - Clock/Enable
 - Data
 - Read/Write
 - Reset
- Μπορούν να χρησιμοποιηθούν ως δύο ανεξάρτητες μνήμες μόνης θύρας με μισό μέγεθος



Dual-port RAM

142



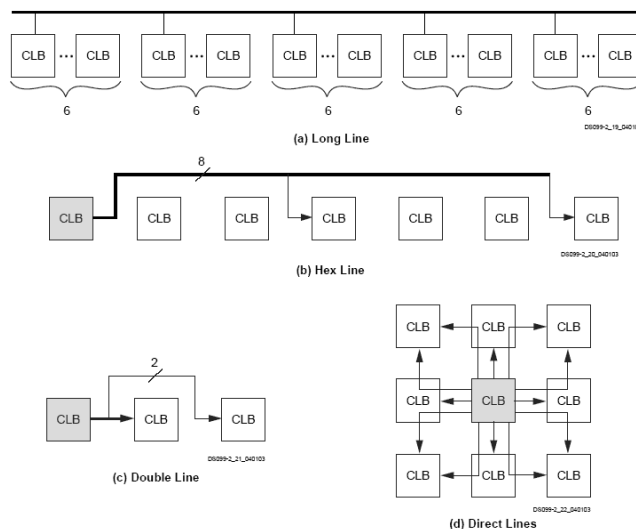
Οικογένεια Xilinx Spartan-3

143

- Προγραμματιζόμενα μπλοκ εισόδου/εξόδου
 - ▣ Input Output Blocks (IOB)
- Διευθυντές ρολογιού
 - ▣ Digital Clock Manager (DCM)
- Προγραμματιζόμενα μπλοκ λογικής
 - ▣ Configurable Logic Blocks (CLB)
- Ευέλικτα μπλοκ μνήμης
 - ▣ BlockRAM
- Προγραμματιζόμενες πηγές δρομολόγησης

Τύποι καλωδίων δρομολόγησης

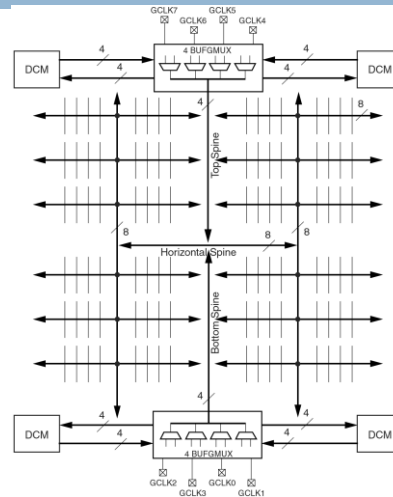
144



Σφαιρική δρομολόγηση

145

- Διανέμει ρολόγια και άλλα σήματα με υψηλό fanout (οδήγηση σημάτων) σε όλη τη συσκευή με χαμηλή απόκλιση (skew)
- Οκτώ σφαιρικά δίκτυα ρολογιού



146

Ενότητα 7

147

- Μέτρηση της απόδοσης μιας FPGA σχεδίασης
 - ▣ Στατική χρονική ανάλυση (static timing analysis)
 - ▣ Χρονικοί περιορισμοί (timing constraints)
 - Πώς εισάγουμε χρονικούς περιορισμούς

Στατική χρονική ανάλυση: γιατί είναι απαραίτητη;

148

- Ακόμα και εάν ένα ψηφιακό σύστημα είναι λογικά σωστό, χρειάζεται να γνωρίζουμε πως λειτουργεί η φυσική του υλοποίηση
 - ▣ Για να σιγουρευτούμε ότι ικανοποιεί κάποιες προδιαγραφές απόδοσης
 - ▣ Για να αξιολογήσουμε την απόδοσή του

Χρονική ανάλυση

149

- Οι πληροφορίες που μας ενδιαφέρουν στη χρονική ανάλυση είναι:
 - ▣ Απαιτήσεις από τα σήματα εισόδου στο κύκλωμα
 - ▣ Εσωτερική απόδοση, μέγιστη συχνότητα ρολογιού
 - ▣ Συμπεριφορά των σημάτων εξόδου από το κύκλωμα
- Η μέγιστη συχνότητα λειτουργίας δηλώνει πόσο γρήγορα μπορεί να λειτουργήσει το κύκλωμα
- Η συμπεριφορά των εισόδων και των εξόδων δηλώνει πως το κύκλωμα αλληλεπιδρά με άλλα κυκλώματα και τον εξωτερικό κόσμο

Χρονικοί περιορισμοί (timing constraints)

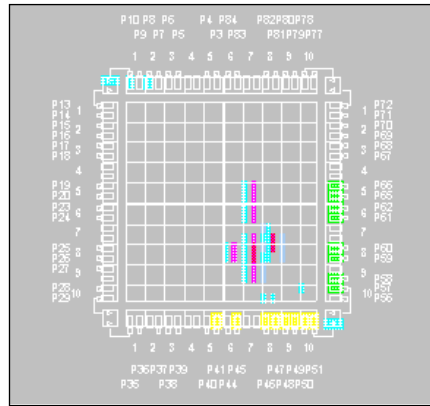
150

- Τι επιδράσεις έχουν οι χρονικοί περιορισμοί στη σχεδίαση ενός FPGA;
 - ▣ Δεν βελτιστοποιούν τη σχεδίαση ή το αποτέλεσμα της σύνθεσης
 - ▣ Τα εργαλεία υλοποίησης δεν επιχειρούν να βρουν τη διάταξη (place & route) που επιτυγχάνει την καλύτερη ταχύτητα
 - ▣ Αντίθετα, επιχειρούν να ικανοποιήσουν τις απαιτήσεις απόδοσης του σχεδιαστή
- Οι απαιτήσεις απόδοσης αποδίδονται με τους χρονικούς περιορισμούς (timing constraints)
 - ▣ Οι χρονικοί περιορισμοί βελτιώνουν την απόδοση της σχεδίασης, τοποθετώντας τη λογική πιο κοντά ώστε να είναι δυνατή η χρησιμοποίηση πιο σύντομων πόρων δρομολόγησης
 - ▣ Οι χρονικοί περιορισμοί περιγράφονται στο εργαλείο Xilinx Constraints Editor

Χωρίς χρονικούς περιορισμούς

151

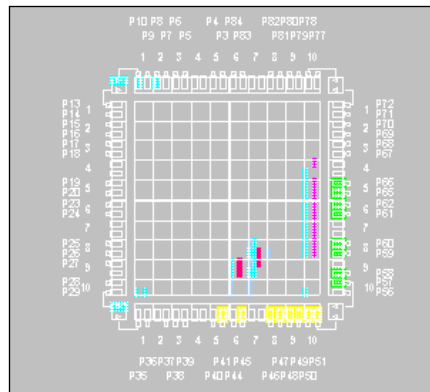
- Η σχεδίαση δεν έχει χρονικούς περιορισμούς (timing constraints) ή αναθέσεις ακροδεκτών (pin assignments)
- Παρατηρήστε την τοποθέτηση της λογικής και των ακροδεκτών
 - ▣ Η λογική έχει ομαδοποιηθεί και τοποθετηθεί σε γειτονικά μπλοκ ώστε να παρέχει υψηλή εσωτερική συχνότητα και να ελαχιστοποιήσει την απόκλιση του ρολογιού
 - ▣ Όμοια, οι ακροδέκτες έχουν ομαδοποιηθεί λογικά



Με χρονικούς περιορισμούς

152

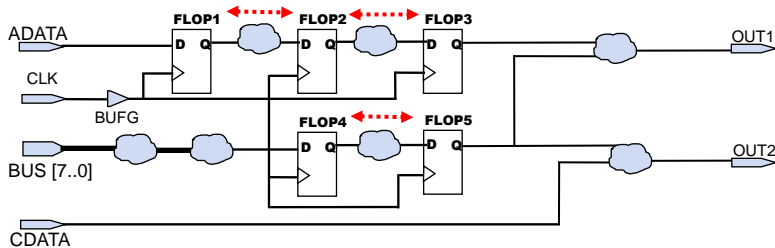
- Η ίδια σχεδίαση με χρονικούς περιορισμούς
- Παρατηρήστε ότι η λογική τοποθετήθηκε πιο κοντά στους ακροδέκτες
 - ▣ Αυτό βελτιώνει τους χρονισμούς και εντός και εκτός του τσιπ και
 - ▣ Επιτυγχάνει καλύτερη απόδοση



Περιορισμοί περιόδου (period constraints)

153

- Οι περιορισμοί περιόδου στοχεύουν τα μονοπάτια μεταξύ σύγχρονων στοιχείων μνήμης



Περιορισμοί περιόδου

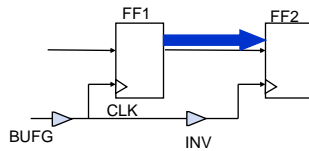
154

- Χρησιμοποιούν την πιο ακριβή χρονική πληροφορία
 - ▣ Απόκλιση ρολογιού (clock skew) μεταξύ των flip-flops πηγής και προορισμού
 - ▣ Σύγχρονα στοιχεία μνήμης με ενεργή ακμή την καθοδική ακμή του ρολογιού
 - ▣ Άνισοι κύκλοι λειτουργίας του ρολογιού (clock duty cycles)
 - ▣ Παραμόρφωση ρολογιού (clock jitter)

Περιορισμοί περιόδου

155

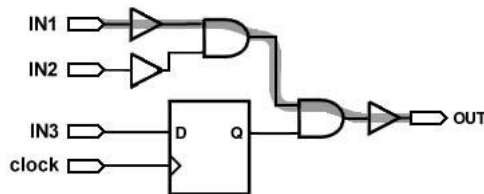
- Παράδειγμα:
 - ▣ 50 % duty cycle στο ρολόι (CLK)
 - ▣ Περιορισμός περιόδου: 10 ns
 - ▣ Το FF2 χρησιμοποιεί την καθοδική ακμή του CLK
 - Η περίοδος του CLK για το μονοπάτι μεταξύ των δύο flip-flops θα περιοριστεί στο $50\% \times 10 \text{ ns} = 5 \text{ ns}$



Περιορισμοί (pad-to-pad constraints)

156

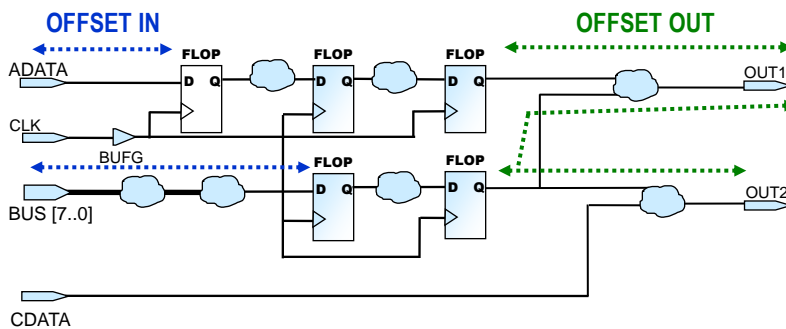
- Καλύπτουν αμιγώς συνδυαστικά μονοπάτια
- Δεν χρησιμοποιούνται συχνά
 - ▣ Τα κυκλώματα σπανίως περιέχουν αμιγώς συνδυαστικά μονοπάτια καθυστέρησης



Περιορισμοί σχετικής απόστασης (offset I/O constraints)

157

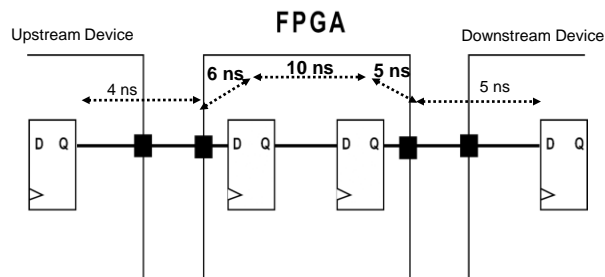
- Καλύπτουν μονοπάτια από τους ακροδέκτες εισόδου (input pads) στα σύγχρονα ακολουθιακά στοιχεία (OFFSET IN) και από τα σύγχρονα ακολουθιακά στοιχεία στους ακροδέκτες εξόδου (output pads) (OFFSET OUT)



Παράδειγμα

158

- Τι περιορισμούς πρέπει να θέσουμε στο Constraints Editor ώστε το κύκλωμα να μπορεί να «τρέξει» στα 100 MHz;



Ενότητα 8

159

- Προβλήματα/λύσεις στη σχεδίαση σύγχρονων ακολουθιακών κυκλωμάτων σε FPGAs
 - ▣ Κυκλώματα οδήγησης του ρολογιού
 - ▣ Κυκλώματα οδήγησης των σημάτων αρχικοποίησης
 - ▣ Κυκλώματα συγχρονισμού για ασύγχρονες εισόδους
 - ▣ Διασταύρωση μεταξύ πεδίων ρολογιού

Γιατί σύγχρονη σχεδίαση;

160

- Τα **σύγχρονα κυκλώματα** είναι πιο αξιόπιστα
 - ▣ Γεγονότα μόνο στις ακμές του ρολογιού, που συμβαίνουν ανά συγκεκριμένα χρονικά διαστήματα
 - ▣ Έξοδοι από μία λογική βαθμίδα έχουν έναν ολόκληρο κύκλο ρολογιού για να διαδοθούν στην επόμενη βαθμίδα
 - ▣ Η απόκλιση (skew) μεταξύ των χρόνων άφιξης των νέων τιμών των σημάτων ρυθμίζεται εντός της ίδιας περιόδου του ρολογιού
- Τα **ασύγχρονα κυκλώματα** είναι λιγότερο αξιόπιστα
 - ▣ Μπορεί να απαιτείται η καθυστέρηση ενός σήματος να είναι συγκεκριμένη, π.χ. 12 ns
 - ▣ Μπορεί να απαιτείται να υπάρχει συγκεκριμένη σχέση μεταξύ πολλαπλών καθυστερήσεων, π.χ. το σήμα DATA να φτάσει 5 ns πριν το σήμα SELECT

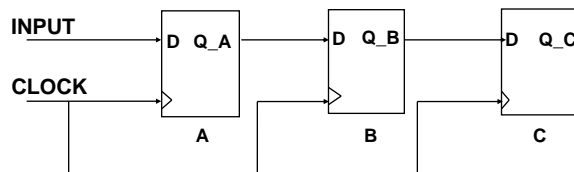
Η Xilinx προτείνει...

161

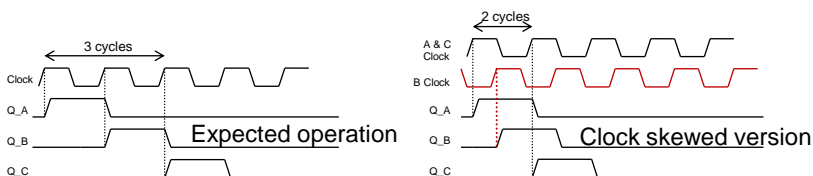
- Για λόγους αξιοπιστίας, σύγχρονη σχεδίαση
- Είναι συνηθισμένο φαινόμενο, ασύγχρονα κυκλώματα να δουλεύουν σωστά σε μια συσκευή και να αποτυγχάνουν ...
 - Σε μια μελλοντική συσκευή FPGA
 - Λόγω της αύξησης της ταχύτητας των ολοκληρωμένων από τη συρρίκνωση της γεωμετρίας
 - Σε μια ταυτόσημη συσκευή
 - Λόγω των φυσιολογικών διακυμάνσεων των καθυστερήσεων εντός του ολοκληρωμένου

Το πρόβλημα της απόκλισης του ρολογιού (clock skew)

162



Ο καταχωρητής ολίσθησης δεν θα δουλέψει λόγω απόκλισης του ρολογιού!



Διανομή του ρολογιού (clock distribution)

163

- Η «ιδανική» διανομή του ρολογιού θα στείλει το ρολόι σε όλα τα σύγχρονα στοιχεία του κυκλώματος με **μηδενική καθυστέρηση** (zero delay) και **μηδενική απόκλιση** (zero skew)
 - ▣ Αυτό είναι που βλέπουμε στην προσομοίωση συνάρτησης (functional simulation)
 - ▣ Δεν αντιπροσωπεύει την πραγματικότητα
- Μερικές σχεδιάσεις μπορεί να κάνουν αποδοτική χρήση της καθυστέρησης και της απόκλισης του ρολογιού
- Οι συσκευές FPGA διαθέτουν **ειδικούς πόρους δρομολόγησης** για σήματα με απαιτήσεις υψηλού fan-out και μικρής απόκλισης, όπως τα ρολόγια
 - ▣ Περιορισμένοι πόροι
 - ▣ Οδηγούνται από “global buffers”

Global buffers

164

- Στις συσκευές FPGA, πρέπει να εκδηλώσεις την επιθυμία σου να χρησιμοποιήσεις αυτούς τους πόρους με τη **δήλωση στιγμιοτύπου** ενός “global buffer” που θα οδηγήσει το σήμα του ρολογιού
 - ▣ Τα περισσότερα εργαλεία σύνθεσης θα εισάγουν αυτόματα ένα **BUFG** στα σήματα ρολογιού
 - Για τα ρολόγια που δηλώνονται στο υψηλότερο επίπεδο σχεδίασης
 - Στα ρολόγια που παράγονται εσωτερικά δεν τοποθετείται αυτόματα BUFG
 - Μερικά εργαλεία σύνθεσης έχουν τη δυνατότητα να αναγνωρίζουν εσωτερικά σήματα που χρησιμοποιούνται ως ρολόγια και να εισάγουν αυτόματα BUFG
- Τα σχηματικά διαγράμματα χρησιμοποιούν ένα συστατικό που ονομάζεται BUFG
- Οι HDL σχεδιάσεις κάνουν χρήση ενός στιγμιοτύπου του BUFG από τη βιβλιοθήκη

Ρολόγια (clocks)

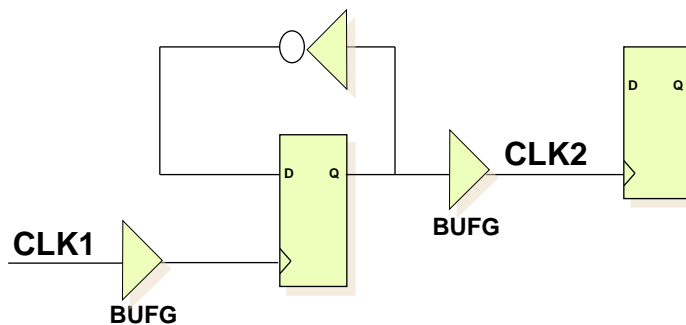
165

- Επειδή οι συσκευές FPGA έχουν περιορισμένους πόρους για τη δρομολόγηση των ρολογιών, οι σχεδιαστές πρέπει να ελαχιστοποιούν τον αριθμό των σημάτων ρολογιού στη σχεδίαση
- **Συμβουλές για τη χρήση ρολογιών**
 - Αποφύγετε την τεχνική “clock gating”
 - Αποφύγετε κυκλώματα όπως οι μετρητές ριπής (ripple counters)
 - Χρησιμοποιείται σήματα επίτρεψης του ρολογιού (clock enables) αντί για διαιρεμένα ρολόγια (divided clocks)
- Με αυτόν τον τρόπο κάνετε λιγότερο πολύπλοκη τη στατική χρονική ανάλυση!

Διαίρεση του ρολογιού

166

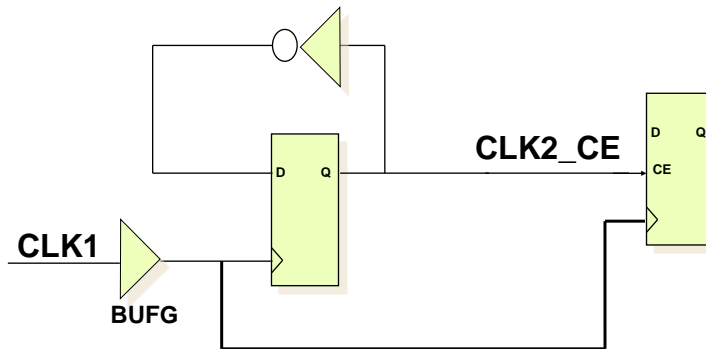
- Εισάγει απόκλιση μεταξύ των ρολογιών CLK1 και CLK2
- Χρησιμοποιεί έναν έξτρα BUFG για να μειώσει την απόκλιση στο ρολόι CLK2



Διαίρεση του ρολογιού

167

- Καμία απόκλιση μεταξύ των ρολογιών των flip-flop



Clock glitches

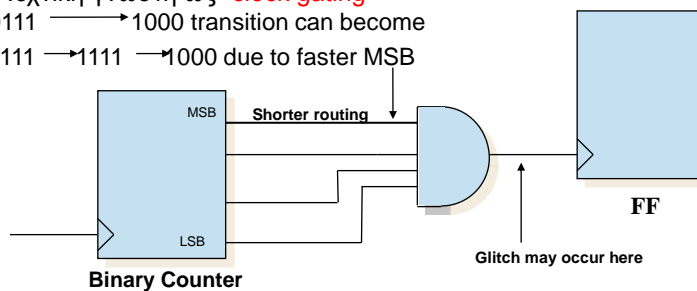
168

- Επειδή τα flip-flop στα σημερινά FPGAs είναι πολύ γρήγορα, μπορεί να ανταποκριθούν σε πολύ μικρούς παλμούς ρολογιού
 - Μην χρησιμοποιείται ως ρολόι ένα σήμα που παράγεται από συνδυαστική λογική

- Τεχνική γνωστή ως “clock gating”

0111 → 1000 transition can become

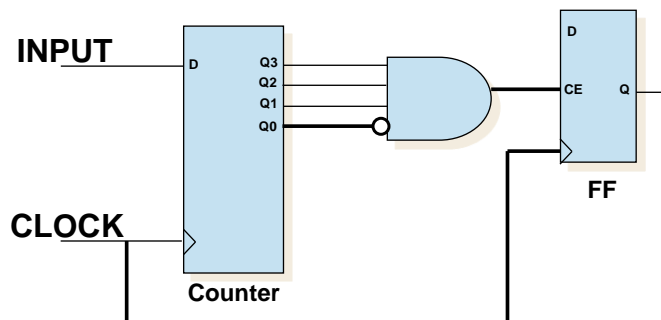
0111 → 1111 → 1000 due to faster MSB



Αποφυγή του clock gating

169

- Αυτό το κύκλωμα παράγει την ίδια συνάρτηση, αλλά χωρίς να δημιουργεί glitches στο σήμα του ρολογιού



VHDL κώδικας για τα σήματα επίτρεψης ρολογιού

170

```
-- FF with clock enable
FF_CE: process (CLK)
begin
  if (CLK'event and CLK = '1') then
    if (ENABLE = '1') then
      Q <= D_IN;
    end if;
  end if;
end process;
```

- Εάν το σήμα ENABLE δεν είναι θύρα στο υψηλότερο επίπεδο σχεδίασης, γράψτε κώδικα για την παραγωγή του ENABLE σε άλλη διεργασία
 - Κάνει τον κώδικα πιο ευανάγνωστο
 - Βοηθάει το εργαλείο σύνθεσης να παράγει καλύτερα αποτελέσματα

Σήματα αρχικοποίησης (reset/set signals)

171

- Τα περισσότερα κυκλώματα χρησιμοποιούν ένα σήμα συγχρονισμού, “reset” ή “set”, για να θέσουν το κύκλωμα σε μια αρχική κατάσταση
- Η αρχική κατάσταση δεν χρειάζεται να αρχικοποιεί όλα τα flip-flop στο 0 ή στο 1
 - ▣ Μπορείτε να θέσετε κάθε flip-flop ξεχωριστά
 - ▣ Μπορείτε να μην αρχικοποιήσετε όλα τα στοιχεία μνήμης
- Τα σήματα αρχικοποίησης μπορεί να είναι
 - ▣ **Σύγχρονα** (synchronous resets)
 - ▣ **Ασύγχρονα** (asynchronous resets)

Synchronous resets

172

- Το σύγχρονο σήμα αρχικοποίησης είναι συγχρονισμένο με το ρολόι του κυκλώματος
 - ▣ Όταν ενεργοποιηθεί, και έρθει η επόμενη ενεργή ακμή του ρολογιού, τότε το flip-flop θα αρχικοποιηθεί
 - ▣ Μπορεί να θεωρηθεί σαν μια ακόμα σύγχρονη είσοδο των flip-flop
 - ▣ Το σύγχρονο σήμα αρχικοποίησης έχει προτεραιότητα έναντι των άλλων σύγχρονων εισόδων ενός flip-flop, π.χ. Clock enable, D input
 - ▣ Έχει τις ίδιες χρονικές απαιτήσεις με τις άλλες σύγχρονες εισόδους ενός flip-flop
- Εάν το σύγχρονο reset προέρχεται από μια εξωτερική πηγή, πρέπει να ικανοποιεί τις απαιτήσεις setup & hold time
- Εάν το σύγχρονο reset προέρχεται από μια εσωτερική πηγή, πρέπει να ικανοποιεί την απαίτηση της περιόδου του ρολογιού

Asynchronous resets

173

- Το ασύγχρονο σήμα αρχικοποίησης δεν είναι συγχρονισμένο με το ρολόι του κυκλώματος
 - ▣ Όταν ενεργοποιηθεί, τότε το flip-flop θα αρχικοποιηθεί αμέσως, χωρίς να απαιτείται ακμή του ρολογιού
 - ▣ Το ασύγχρονο σήμα αρχικοποίησης έχει προτεραιότητα έναντι όλων των άλλων εισόδων ενός flip-flop, ακόμα και του ρολογιού
- Σε αντίθεση με ένα σύγχρονο reset, ένα ασύγχρονο reset μπορεί να δημιουργήσει προβλήματα όταν απενεργοποιηθεί
 - ▣ Μπορεί να συμβεί οποιαδήποτε χρονική στιγμή, ακόμα και κοντά στις ακμές του ρολογιού
 - ▣ Απόκλιση στη διανομή του σήματος reset, μπορεί να οδηγήσει διαφορετικά τμήματα του κυκλώματος να «ξυπνήσουν» σε διαφορετικές χρονικές στιγμές
 - Η σχεδίαση μπορεί να βρεθεί σε κάποια κατάσταση που δεν προβλέπεται

174

```
entity hang_yourself is
    port ( clk, rst : in bit; detonate_warhead: out bit);
end entity;

architecture beh of hang_yourself is
    signal flop1, flop2 : bit;
begin

    P1: process (rst, clk)
        begin
            if (rst = '1') then flop1 <= '0';
            elsif (clk'event and clk = '1') then
                flop1 <= !flop1;
            end if;
        end process;

    P2: process (rst, clk)
        begin
            if (rst = '1') then flop2 <= '0';
            elsif (clk'event and clk = '1') then
                flop2 <= !flop2;
            end if;
        end process;

    P3: process (rst, clk)
        begin
            if (rst = '1') then detonate_warhead <= '0';
            elsif (clk'event and clk = '1') then
                detonate_warhead <= flop1 ^ flop2;
            end if;
        end process;

end architecture;
```

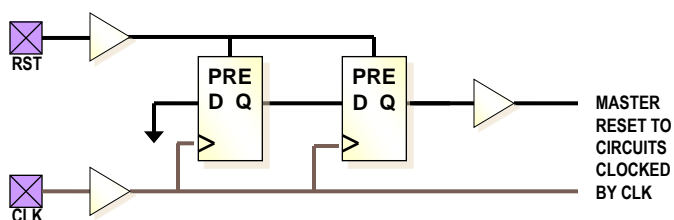
**Πρόβλημα με
τα ασύγχρονα
resets**



Λύση στο πρόβλημα

175

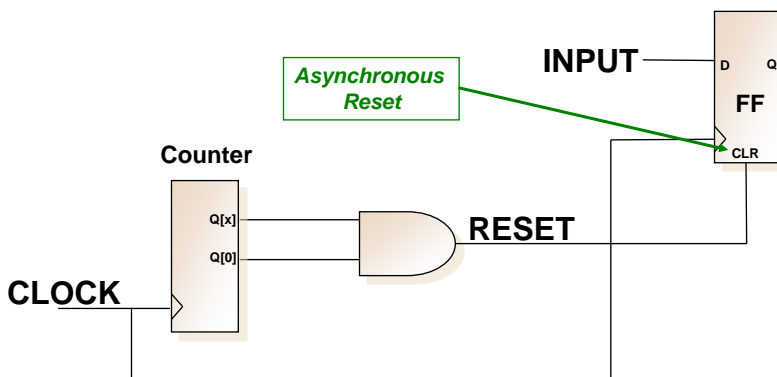
- Κύκλωμα για την οδήγηση των ασύγχρονων resets:
 - ▣ Το σήμα reset **ενεργοποιείται ασύγχρονα**
 - Οδηγεί το κύκλωμα άμεσα στην κατάσταση αρχικοποίησης
 - ▣ Το σήμα reset **απενεργοποιείται σύγχρονα**
 - Επιτρέπει τη «σωστή» χρονική ανάλυση του κυκλώματος



Αποφυγή των set/reset glitches

176

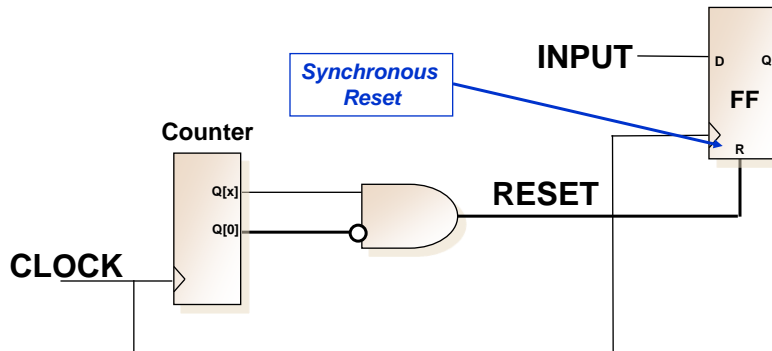
- Glitches σε ασύγχρονα σήματα set/reset μπορεί να οδηγήσουν σε λανθασμένη συμπεριφορά του κυκλώματος



Αποφυγή των set/reset glitches

177

- Μετατροπή σε σύγχρονα σήματα set/reset όπου είναι δυνατόν



VHDL κώδικας για σύγχρονα/ασύγχρονα set/reset

178

-- FF with **asynchronous** reset/set and clock enable

```
FF_SR_SS_CE:
process (RESET, SET, CLK)
begin
    if (RESET = '1') then
        Q <= '0';
    elsif (SET = '1') then
        Q <= '1';
    elsif (CLK'event and CLK = '1') then
        if (ENABLE = '1') then
            Q <= D_IN;
        end if;
    end if;
end process;
```

-- FF with **synchronous** reset/set and clock enable

```
FF_AR_AS_CE:
process (CLK)
begin
    if (CLK'event and CLK = '1') then
        if (RESET = '1') then
            Q <= '0';
        elsif (SET = '1') then
            Q <= '1';
        elsif (ENABLE = '1') then
            Q <= D_IN;
        end if;
    end if;
end process;
```

- Σειρά προτεραιότητας των σημάτων ελέγχου: Reset, Set, Enable
 - Διατηρήστε αυτήν τη σειρά στον κώδικα

Xilinx FPGA Resets

179

- Κατά τη διάρκεια της διαμόρφωσής του, το FPGA έχει μια μη-ολοκληρωμένη περιγραφή της λογικής
 - ▣ Η κατάσταση του κυκλώματος μπορεί να μεταβληθεί
 - ▣ Τα σήματα εξόδου μπορεί να οδηγηθούν λανθασμένα
- Στα Xilinx FPGAs υπάρχουν **δύο σήματα αρχικοποίησης**
 - ▣ **GSR**, global set/reset
 - ▣ **GTS**, global three state

Xilinx FPGA Resets

180

- Έως ότου ολοκληρωθεί ο προγραμματισμός, το FPGA διατηρεί τα σήματα GSR και GTS ενεργοποιημένα
 - ▣ Όλα τα flip flops διατηρούνται στην αρχική τους κατάσταση (0 ή 1) από το GSR
 - ▣ Όλες οι έξοδοι διατηρούνται σε κατάσταση High-Z από το GTS
- Όταν ολοκληρωθεί ο προγραμματισμός, τα σήματα GSR και GTS απενεργοποιούνται, και το κύκλωμα αρχίζει να λειτουργεί κανονικά

Xilinx FPGA Resets

181

- Κατά τη διάρκεια της κανονικής λειτουργίας του κυκλώματος, επιτρέπεται ο χρήστης να οδηγήσει τα σήματα GSR και GTS
 - ▣ Η ενεργοποίηση του GSR θα αρχικοποιήσει όλα τα flip-flop και τα latches
 - ▣ Η ενεργοποίηση του GTS θα θέσει σε High-Z όλες τις εξόδους
 - ▣ Ο έλεγχος των σημάτων GSR και GTS γίνεται μέσω του συστατικού της βιβλιοθήκης **STARTUP**
 - Μπορείτε να χρησιμοποιήσετε το συστατικό, είτε σε σχεδιάσεις HDL, είτε σε σχηματικά διαγράμματα

Κυκλώματα συγχρονισμού (synchronization circuits)

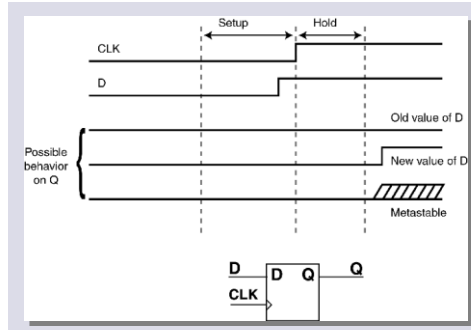
182

- Τι είναι ένα **κύκλωμα συγχρονισμού**;
 - ▣ Λαμβάνει (δειγματοληπτεί) ένα ασύγχρονο σήμα εισόδου και παράγει μια έξοδο συγχρονισμένη με την ακμή του ρολογιού
- Γιατί χρειαζόμαστε κυκλώματα συγχρονισμού;
 - ▣ Για να αποφύγουμε παραβιάσεις των χρόνων setup και hold
 - ▣ Για να εξασφαλίσουμε ένα πιο αξιόπιστο κύκλωμα
- Πού χρειαζόμαστε κυκλώματα συγχρονισμού;
 - ▣ Σήματα που διασταυρώνονται μεταξύ ασυσχέτιστων πεδίων ρολογιού (unrelated clock domains)
 - ▣ Είσοδοι του τσιπ που είναι ασύγχρονες
 - Π.χ. buttons, switches, κτλ.

Παραβιάσεις των χρόνων setup και hold

183

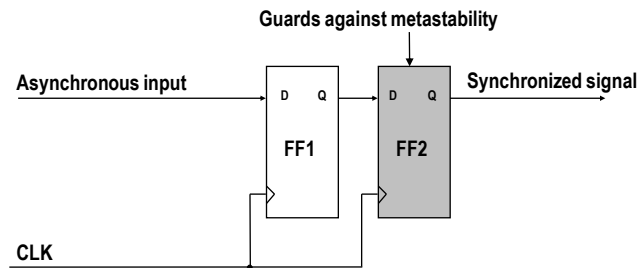
- Παραβίαση συμβαίνει όταν η είσοδος του flip-flop αλλάζει πολύ «κοντά» στην ακμή του ρολογιού
- Τρία πιθανά αποτελέσματα:
 - ▣ Το flip-flop δειγματοληπτεί την «παλιά» τιμή του D
 - ▣ Το flip-flop δειγματοληπτεί τη «νέα» τιμή του D
 - ▣ Η έξοδος του flip-flop γίνεται μετασταθής (metastable)



Κύκλωμα συγχρονισμού 1

184

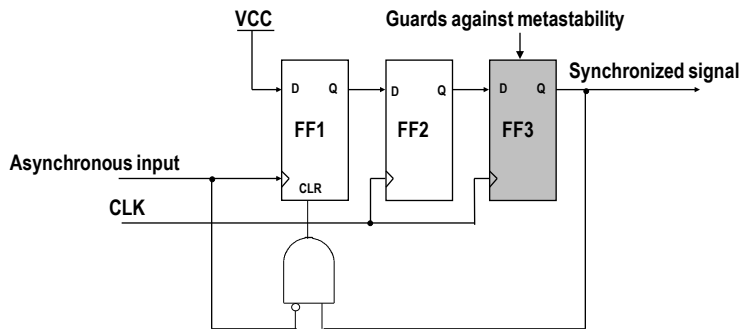
- Το κύκλωμα είναι ένας απλός **2-bit καταχωρητής ολίσθησης** (shift register)
- Χρησιμοποιείται όταν οι παλμοί εισόδου έχουν εύρος **τουλάχιστον** μια περίοδο ρολογιού
- Τα «έξτρα» flip-flop προφυλάσσουν από τη μεταστάθεια



Κύκλωμα συγχρονισμού 2

185

- Χρησιμοποιείται όταν οι παλμοί εισόδου έχουν εύρος **μικρότερο** από μια περίοδο ρολογιού
 - ▣ Το flip-flop 1 «πιάνει» μικρούς παλμούς



Κύκλωμα συγχρονισμού 2

186

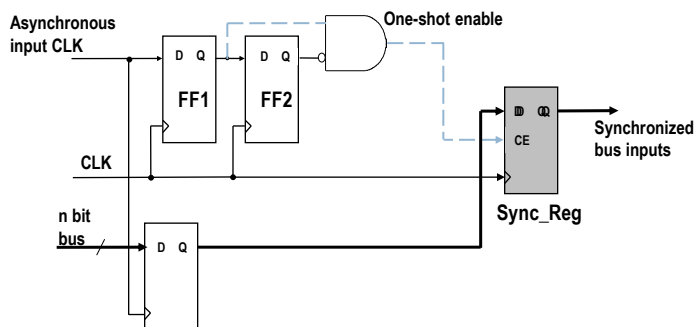
- Προβλήματα του κυκλώματος συγχρονισμού 2:
 - ▣ Εάν πολλαπλοί «μικροί» παλμοί συμβούν εντός ενός χρονικού διαστήματος 3 κύκλων ρολογιού, μόνον ο πρώτος παλμός θα γίνει αντιληπτός από το κύκλωμα
 - Αυτό είναι πρόβλημα όταν από ένα πεδίο γρήγορου ρολογιού τροφοδοτούμε δεδομένα προς ένα πεδίο αργού ρολογιού
 - ▣ Χρησιμοποιεί την είσοδο σαν ρολόι (χρήση global buffer?),
- Λύση:
 - ▣ Χρήση ενός γρηγορότερου ρολογιού για να συγχρονίσει την είσοδο
 - ▣ Χρήση του ίδιου ρολογιού, διαιρεμένου, στο υπόλοιπο κύκλωμα
 - ▣ Π.χ. χρήση του DCM

Συγχρονισμός αγωγού

187

- Όταν οι παλμοί εισόδου έχουν εύρος **τουλάχιστον** μια περίοδο ρολογιού

Circuit 1

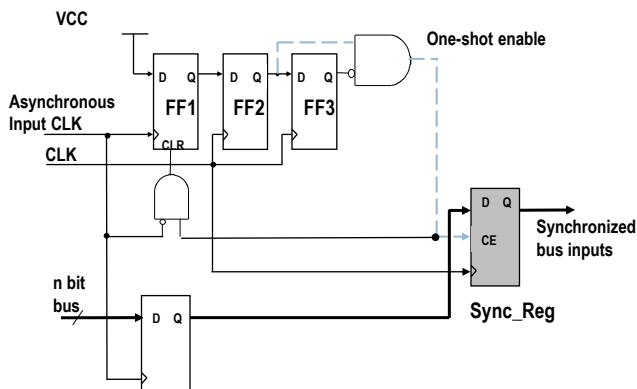


Συγχρονισμός αγωγού

188

- Όταν οι παλμοί εισόδου έχουν εύρος μικρότερο από μια περίοδο ρολογιού

Circuit 2



Ασύγχρονες εισοδοι

189

- Τα κυκλώματα συγχρονισμού προσθέτουν **καθυστέρηση** (latency)
- Τα κυκλώματα συγχρονισμού **δεν εγγυώνται** την εξαφάνιση της μεταστάθειας
- Όταν ένα σήμα εισέλθει στο τσιπ
 - ▣ Να το συγχρονίσετε πρώτα και μετά να το διανείμετε (fan out), όπως απαιτείται
 - ▣ Μην το διανείμετε πρώτα, και μετά να συγχρονίσετε τους πολλαπλούς κλάδους

Πεδία ρολογιού (clock domains)

190

- Ένα **πεδίο ρολογιού** είναι ένα σύνολο λογικών στοιχείων και σημάτων που είναι συγχρονισμένα σε ένα ρολόι
- **Σχεδιαστική συμβουλή:**
 - ▣ Προσπαθήστε να σχεδιάσετε με ένα μόνο πεδίο ρολογιού
 - ▣ Σε πολλές σχεδιάσεις αυτό δεν είναι εφικτό
- Γιατί να έχουμε πολλαπλά πεδία ρολογιού;
 - ▣ Ανεξάρτητα (υπο)συστήματα με διαφορετικά ρολόγια αναφοράς, χρειάζεται να ανταλλάξουν πληροφορία
 - ▣ Μη πρακτικό να διανείμουμε ή να χρησιμοποιήσουμε ένα ρολόι αναφοράς

Πεδία ρολογιού (clock domains)

191

- Πώς μπορεί να σχετίζονται τα ρολόγια σε δυο πεδία;
 - ▣ **Σύγχρονα (synchronous)** - (degenerate case)
 - Ίδια συχνότητα
 - Μηδενική διαφορά φάσης
 - ▣ **Σύγχρονα (synchronous)** - (derived case)
 - Συχνότητες σχετικές σε ένα κοινό ρολόι αναφοράς
 - Διαφορά φάσης συνάρτηση του χρόνου
 - Παράδειγμα: Multiplied or divided clock from DLL
 - ▣ **Mesochronous**
 - Ίδια συχνότητα
 - Σταθερή διαφορά φάσης
 - Παράδειγμα: Phase shifted clock from DLL

Πεδία ρολογιού (clock domains)

192

- Πώς μπορεί να σχετίζονται τα ρολόγια σε δυο πεδία;
 - ▣ **Plesiochronous**
 - Διαφορετικές συχνότητες, ίδια ονομαστική τιμή
 - Διαφορά φάσης ποικίλλει ελαφρώς
 - Παράδειγμα: Two oscillators, both marked 1.0 MHz
 - ▣ **Ασύγχρονα (Asynchronous)**
 - Διαφορετικές συχνότητες
 - Αυθαίρετη διαφορά φάσης
 - Παράδειγμα: Two clocks of unknown relationship

Διασταύρωση πεδίων ρολογιού

193

- Για ασύγχρονα πεδία ρολογιού:
 - ▣ Για ένα απλό σήμα, χρησιμοποιήστε το απλό κύκλωμα συγχρονισμού των δύο flip-flop
 - ▣ Για σήματα πολλών bit, απλά συγχρονίζοντας κάθε bit δεν είναι αρκετό, γιατί κάθε στιγμιότυπο του κυκλώματος συγχρονισμού μπορεί να λειτουργήσει σε διαφορετικές χρονικές στιγμές
 - Δεν μπορούμε να γνωρίζουμε πότε όλα τα bit του σήματος είναι συγχρονισμένα, εκτός και αν περιμένουμε για πολύ...
 - Χρήση χειραψίας (four phase ή two phase handshaking) – απλό σημείο συγχρονισμού

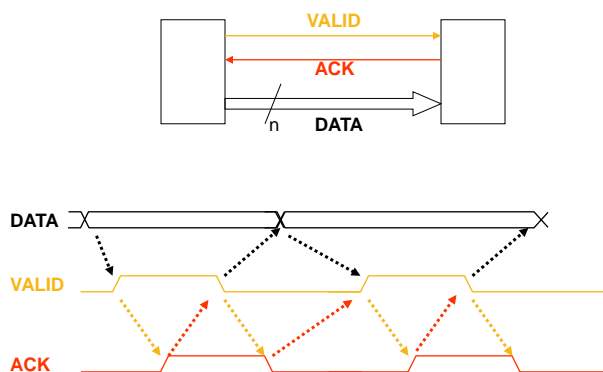
Διασταύρωση πεδίων ρολογιού

194

- Για ασύγχρονα πεδία ρολογιού:
 - ▣ **Four phase handshaking**
(RTZ, Return-to-zero, level based flags)
 - Το πεδίο πηγής παρέχει τα DATA και ενεργοποιεί το σήμα VALID
 - Το πεδίο προορισμού βλέπει το συγχρονισμένο σήμα VALID ενεργοποιημένο και παίρνει τα DATA, και μετά ενεργοποιεί το σήμα ACK
 - Το πεδίο πηγής βλέπει το συγχρονισμένο σήμα ACK ενεργοποιημένο και απενεργοποιεί το σήμα VALID
 - Το πεδίο προορισμού βλέπει το συγχρονισμένο σήμα VALID απενεργοποιημένο και απενεργοποιεί το σήμα ACK
 - Η διαδικασία μπορεί να επαναληφθεί...

Four phase handshake

195



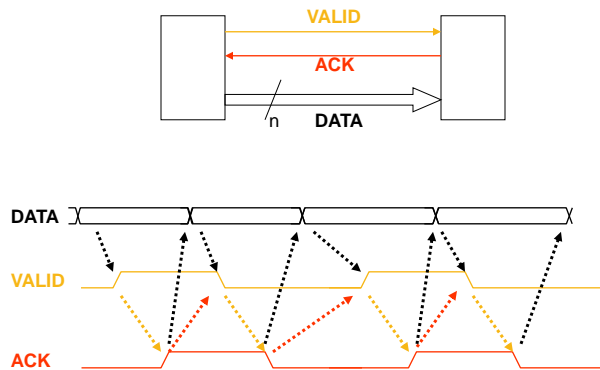
Διασταύρωση πεδίων ρολογιού

196

- Για ασύγχρονα πεδία ρολογιού:
 - ▣ **Two phase handshaking**
(NRZ, Non-return-to-zero, transition flags)
 - Το πεδίο πηγής παρέχει τα DATA και αλλάζει το σήμα VALID
 - Το πεδίο προορισμού βλέπει το συγχρονισμένο σήμα VALID αλλαγμένο και παίρνει τα DATA, και μετά αλλάζει το σήμα ACK
 - Το πεδίο πηγής βλέπει το συγχρονισμένο σήμα ACK αλλαγμένο
 - Η διαδικασία μπορεί να επαναληφθεί...

Two phase handshake

197



Διασπαύρωση πεδίων ρολογιού

198

- Για ασύγχρονα πεδία ρολογιού με ανταλλαγή μεγάλου όγκου πληροφοριών (bulk transfer) χρησιμοποιήστε μια **ασύγχρονη FIFO**:
 - Συνήθως υλοποιείται με μια dual-port memory
 - Στο πεδίο πηγής (write), τα δεδομένα μπορούν να εγγραφούν στη FIFO, όσο η FIFO δεν είναι γεμάτη (FULL)
 - Στο πεδίο προορισμού (read), τα δεδομένα μπορούν να διαβαστούν από τη FIFO, όσο η FIFO δεν είναι άδεια (EMPTY)

Κύκλωμα συγχρονισμού με ασύγχρονη FIFO

199

