

1

ΕΤΕΡΟΓΕΝΗ ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

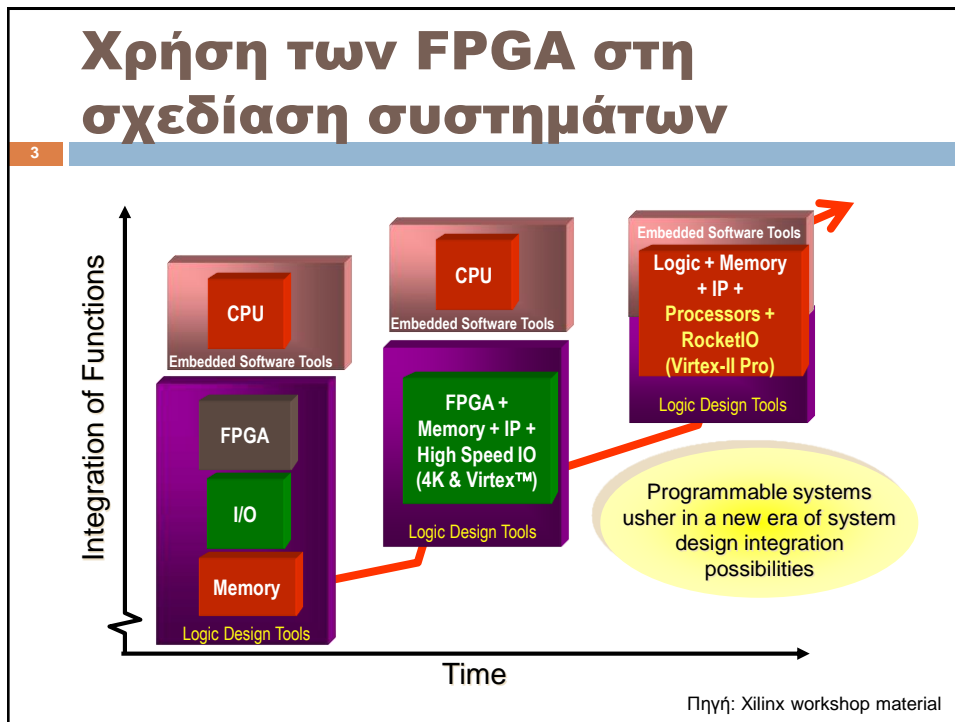
ΣΧΕΔΙΑΣΗ ΣΥΣΤΗΜΑΤΩΝ ΜΕ ΧΡΗΣΗ FPGA

Επικ.Καθηγητής Μιχάλης Ψαράκης

Ενότητα 1

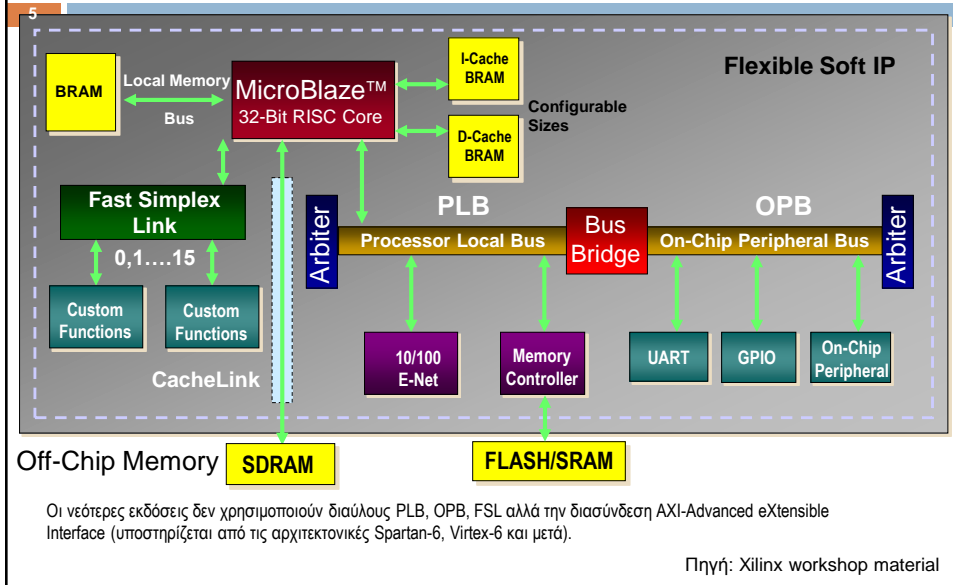
2

- Σχεδίαση ενσωματωμένου συστήματος σε FPGA
 - ▣ Μέσα από τα εργαλεία EDK (embedded development kit) της Xilinx
 - XPS: Xilinx platform studio
 - SDK: Software development kit
 - ▣ Με χρήση soft processor (Xilinx Microblaze)



- ## Σχεδίαση ενσωματωμένου συστήματος σε FPGA
- 4
- Η σχεδίαση ενός ενσωματωμένου συστήματος σε ένα FPGA εμπεριέχει τα εξής:
 - ▣ Ανάπτυξη του ενσωματωμένου συστήματος (επεξεργαστής, δίαυλοι, περιφερειακά, μνήμες)
 - ▣ Ανάπτυξη οδηγών (drivers) για τις περιφερειακές συσκευές
 - ▣ Ανάπτυξη του ενσωματωμένου λογισμικού
 - Interrupt service routines
 - Operating System (OS) ή Real Time Operating System (RTOS)
 - Λογισμικό εφαρμογών

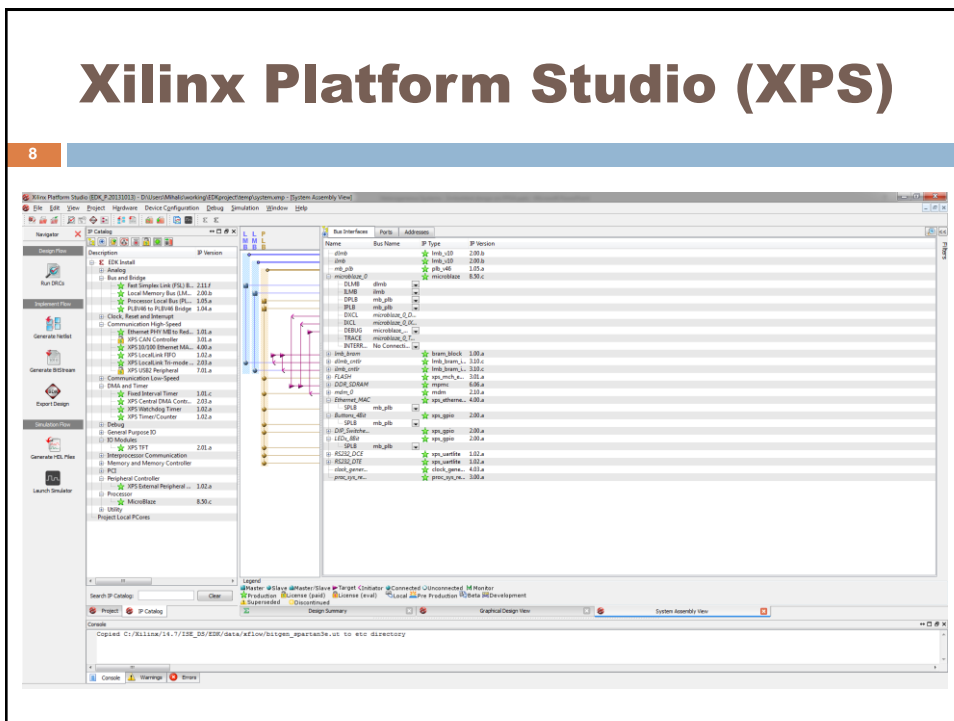
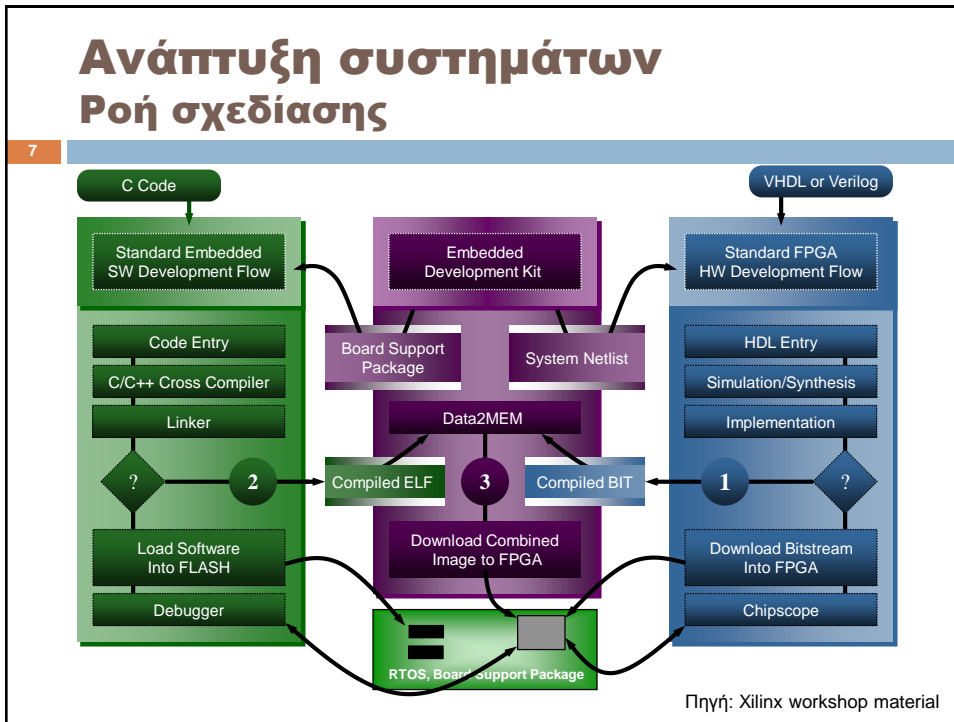
Ενσωματωμένο σύστημα με βάση τον επεξεργαστή MicroBlaze (soft core)



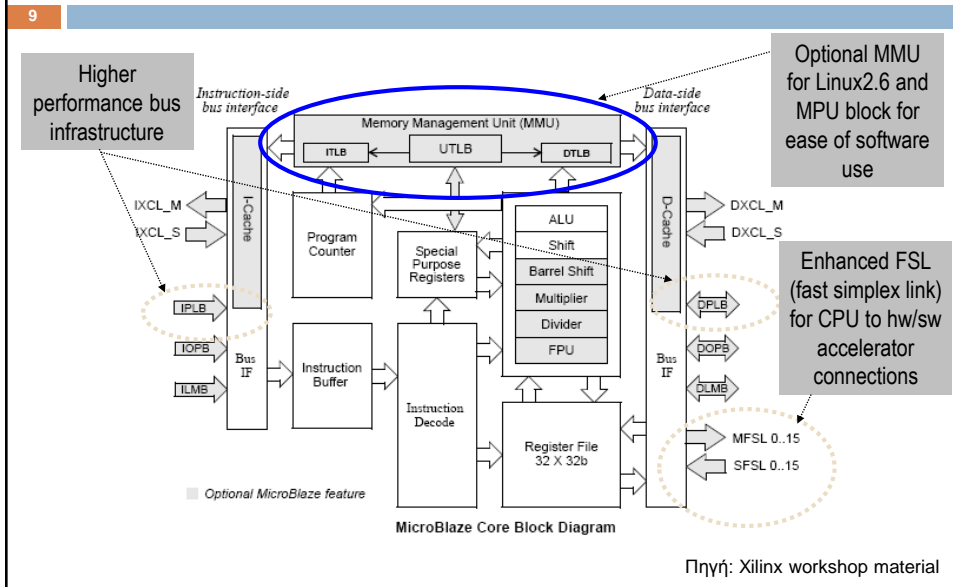
Embedded Development Kit (EDK)

6

- Θα το χρησιμοποιήσετε στην εργασία FPGA-2
- Παρέχει μια πλατφόρμα για την σχεδίαση ενσωματωμένων συστημάτων
 - Παρέχει IP (intellectual property) cores και εργαλεία για την σχεδίαση ενσωματωμένων συστημάτων με βάση τους επεξεργαστές IBM PowerPC™ (hard core) και MicroBlaze™ (soft core)



Σχηματικό διάγραμμα του MicroBlaze



Χαρακτηριστικά του Microblaze

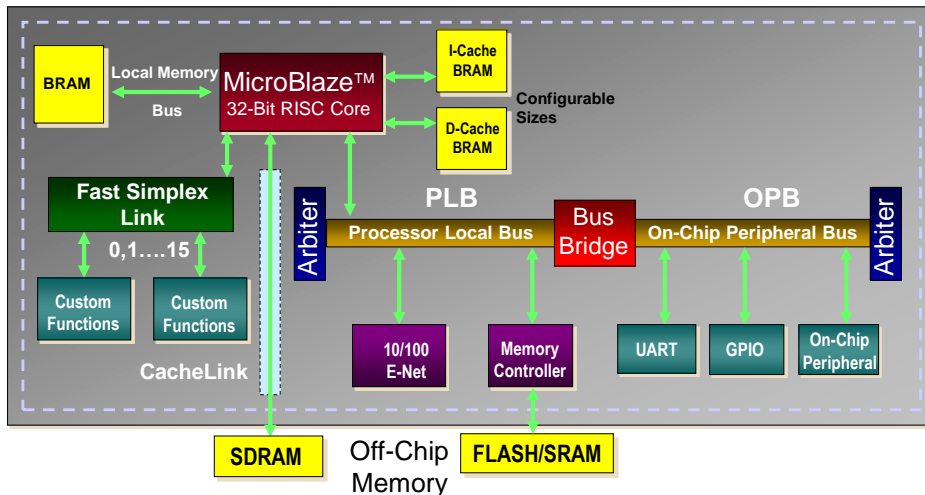
10

- 32-bit RISC επεξεργαστής
 - Μηχανισμός διοχέτευσης μονής έκδοσης (Single-Issue pipeline)
 - 3 (area optimized) ή 5 σταδίων (performance optimised)
 - Προγραμματιζόμενες κρυφές μνήμες εντολών και δεδομένων (instruction and data caches)
 - Direct mapped
 - Μονάδα διαχείρισης και προστασίας μνήμης (προαιρετικά)
 - Απαιτείται για την ενσωμάτωση λειτουργικού Linux OS
 - Μονάδα κινητής υποδιαστολής (Floating-point unit - FPU)
 - Αριθμητικές μονάδες:
 - Ολισθητής (barrel shifter), πολλαπλασιαστής 32x32, διαιρέτης
 - Μονάδα αποσφαλμάτωσης (Hardware Debug and Trace Module)
- Δυνατότητα για σχεδίαση πολυπύρηνου συστήματος
 - Mailbox: επιτρέπει το πέρασμα μηνυμάτων (message passing) μεταξύ 2 πυρήνων
 - Processor ID register

Πηγή: Xilinx workshop material

Διασυνδέσεις (interfaces)

11



Πηγή: Xilinx workshop material

Διασυνδέσεις (interfaces)

12

- **Processor Local Bus (PLB)**
 - ▣ Για συσκευές με μεγάλο εύρος ζώνης (bandwidth), π.χ. Ethernet core
 - ▣ 32-bit address, 32/64/128-bit data bus
 - ▣ Τοπολογία shared bus ή point-to-point
 - ▣ Πολύπλοκη σχεδίαση (priority-based arbitration, address pipeline support)
- **On-Chip Peripheral Bus (OPB)**
 - ▣ Για συσκευές με μικρότερο εύρος ζώνης, π.χ. UART, GPIO
 - ▣ Μικρότερη πολυπλοκότητα, αποσυνδέει τις «αργές» συσκευές από τον «γρήγορο» διάλο
 - ▣ 32-bit address, 32-bit data bus
 - ▣ Τοπολογία shared bus

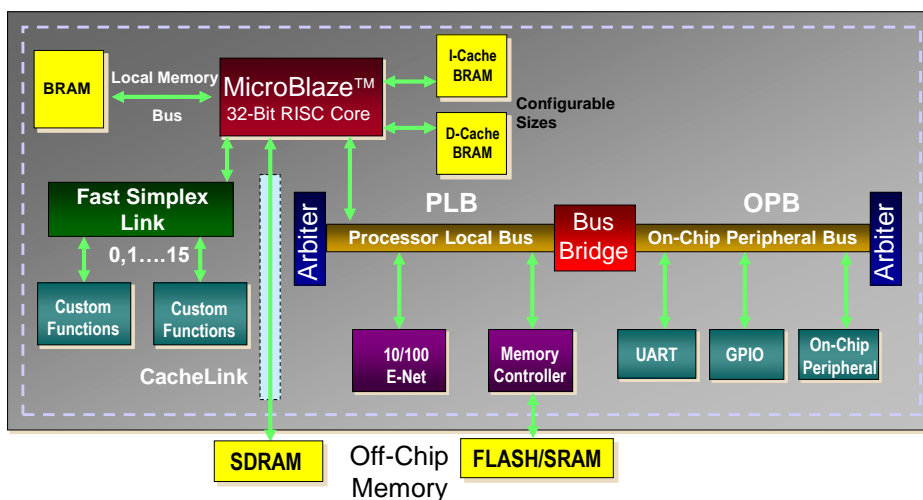
Διασυνδέσεις (interfaces)

13

- Local Memory Bus (LMB)
 - ▣ Παρέχει γρήγορη πρόσβαση στις ενσωματωμένες μνήμες (Block RAM)
- Fast Simplex Links (FSL)
 - ▣ Για γρήγορη πρόσβαση, 600 MHz λειτουργία
 - ▣ MicroBlaze™ C και assembly εντολές για εύκολη προσπέλαση
- Xilinx Cache Link (XCL)
 - ▣ Παρέχει υψηλή απόδοση στο σύστημα μνήμης
 - ▣ Είναι διαθέσιμο μόνο όταν ενεργοποιηθούν οι κρυφές μνήμες

Περιφερειακά

14



Πηγή: Xilinx workshop material

Xilinx IP cores

15

- Παρέχει μια μεγάλη γκάμα από IP cores:
 - Επικοινωνία: High-Speed
 - 10/100 Ethernet MAC, CAN controller, Flexray, USB
 - Επικοινωνία: Low-Speed
 - Serial Peripheral Interface, I2C Interface, UART
 - DMA και χρονομετρητές (timers)
 - Fixed interval timer, watchdog timer, DMA controller
 - Memory Controllers
 - Block RAM, DDR/DDR2/SDRAM, SRAM/Flash
 - General Purpose I/O
 - General Purpose I/O (GPIO)

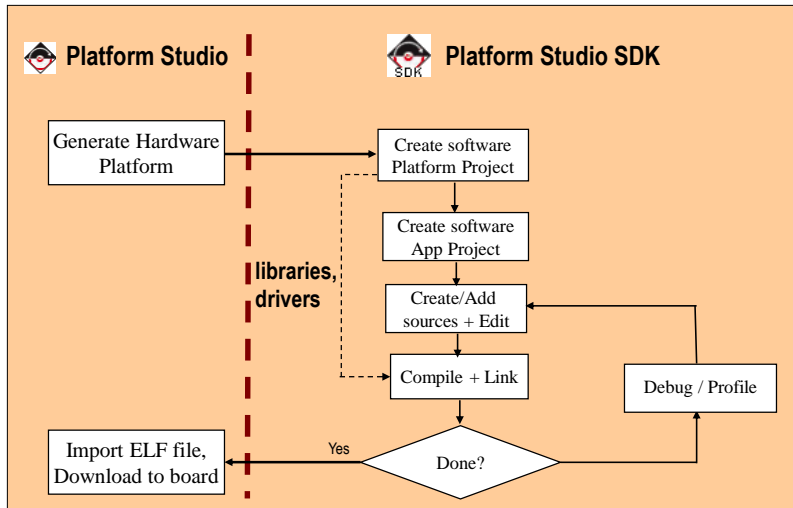
Περιφερειακό του χρήστη

16

- Ο χρήστης μπορεί να δημιουργήσει το δικό του περιφερειακό (με την βοήθεια ενός wizard) και να το εισάγει στην σχεδίαση
- Το εργαλείο παράγει την απαιτούμενη λογική για την διασύνδεση του περιφερειακού με τον επεξεργαστή
 - Ο χρήστης μπορεί να επιλέξει τύπο και χαρακτηριστικά διασύνδεσης, εύρος δεδομένων και διευθύνσεων
 - Και να προσθέσει στην συνέχεια την λογική του περιφερειακού

Ανάπτυξη λογισμικού Ροή σχεδίασης

17



Πηγή: Xilinx workshop material

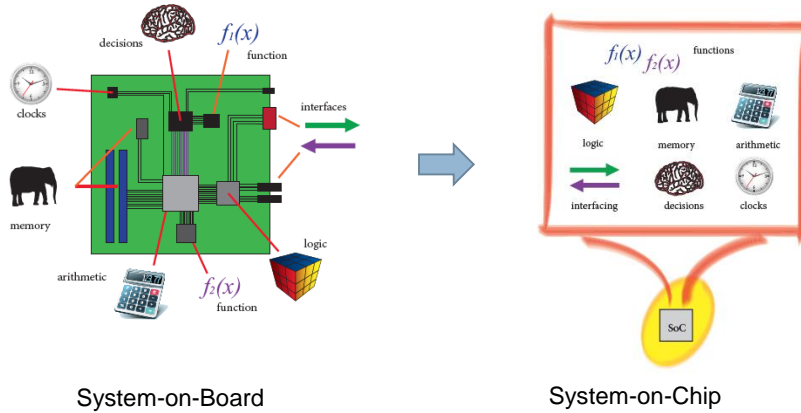
Ενότητα 2

18

- Σχεδίαση ενσωματωμένου συστήματος στην πλατφόρμα Zynq
 - ▣ Αρχιτεκτονική Zynq AP (All Programmable) SoC

System-on-Chip (SoC)

19

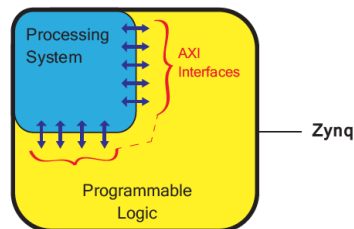


Πηγή: The Zynq Book

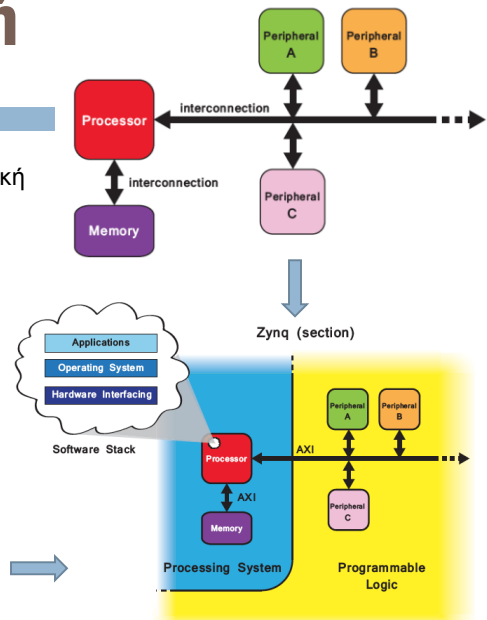
Αρχιτεκτονική Zynq

20

Τυπική αρχιτεκτονική ενός SoC



Απλουστευμένο μοντέλο Zynq

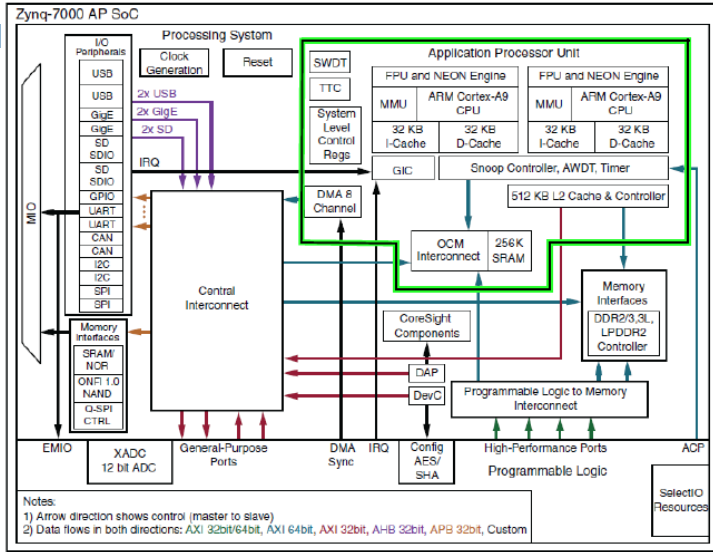


Απεικόνιση SoC στην Zynq αρχιτεκτονική

Πηγή: The Zynq Book

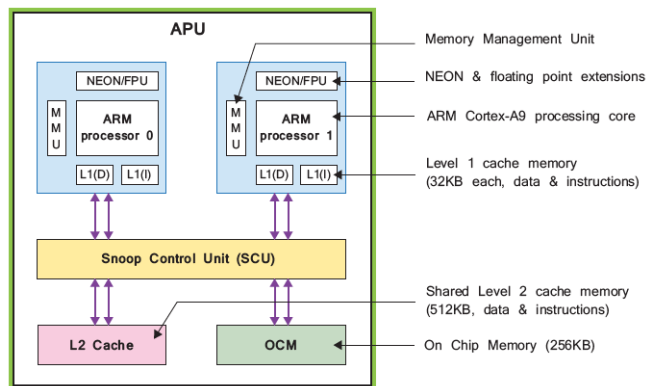
Zynq processing system (PS)

21



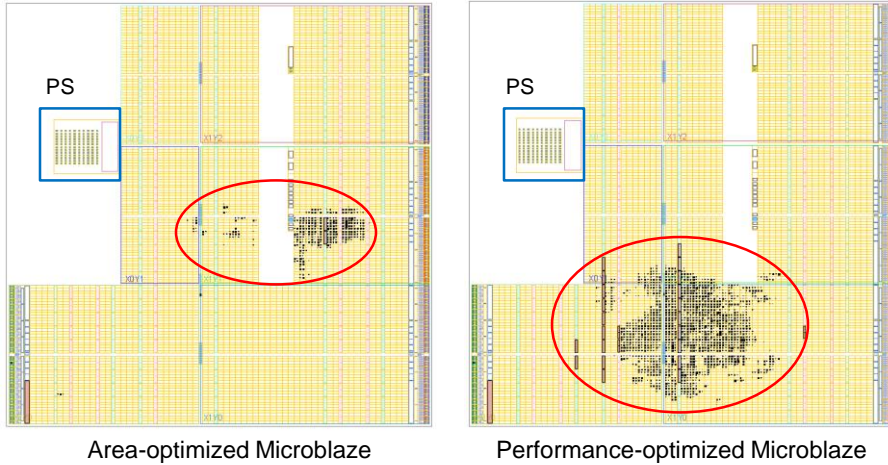
Application processing unit (APU)

22



Zynq vs. FPGA

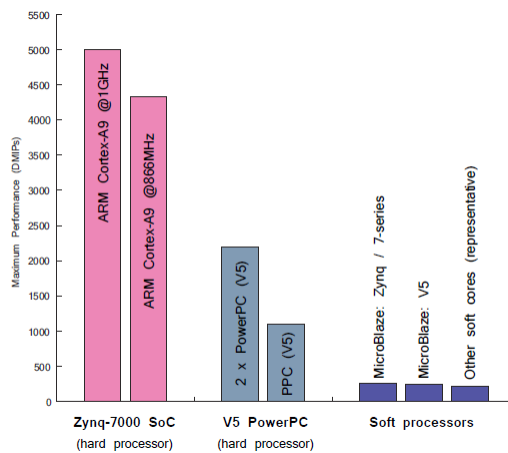
23



Πηγή: The Zynq Book

Hard vs. soft processor

24

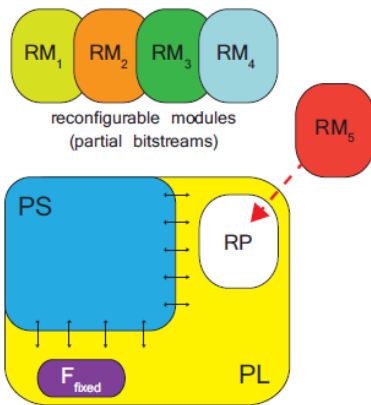


Σύγκριση απόδοσης hard & soft processors

Πηγή: The Zynq Book

Dynamic partial reconfiguration (DPR)

25

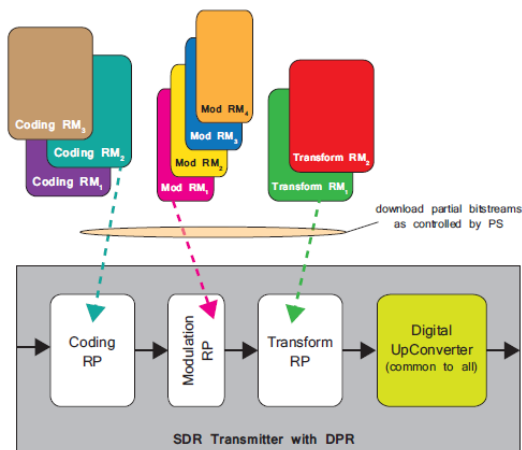


- Πολλαπλά reconfigurable partitions (RP)
- Κάθε RP μπορεί να έχει πολλαπλά reconfigurable modules (RM)
 - ▣ Μέγεθος RP ώστε να χωράει και το μεγαλύτερο RM
 - ▣ Precompiled bitstreams

Πηγή: The Zynq Book

DPR: Παράδειγμα

26

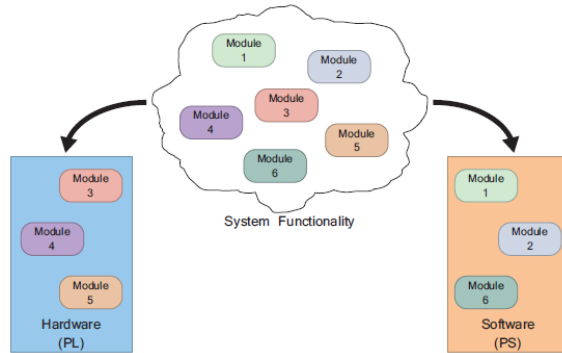


Ευέλικτη αρχιτεκτονική για
Software-Defined Radio (SDR)

Πηγή: The Zynq Book

Hardware/software partitioning

27



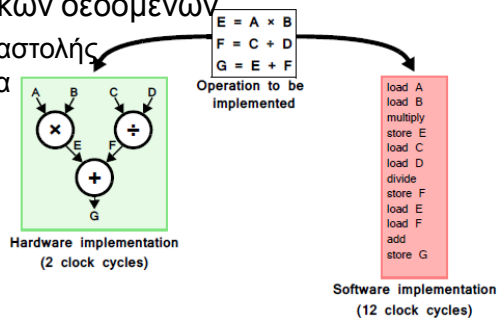
Πηγή: The Zynq Book

Hardware/software partitioning

28

Ζητήματα που καθορίζουν την διαμέριση μεταξύ υλικού και λογισμικού

- Το υλικό ευνοεί τις εφαρμογές με υψηλό βαθμό παραλληλίας
- Αναπαράσταση αριθμητικών δεδομένων
 - Αριθμητική κινητής υποδιαστολής υλοποιείται αποδοτικότερα σε επεξεργαστές
- Επιβάρυνση επικοινωνίας



Πηγή: The Zynq Book

Ροή σχεδίασης με Zynq

29

- Χρήση των εργαλείων ISE/EDK (XPS, SDK)
 - ▣ Παρόμοια με την σχεδίαση με Microblaze
- Χρήση των εργαλείων Vivado
 - ▣ High-level synthesis (HLS)

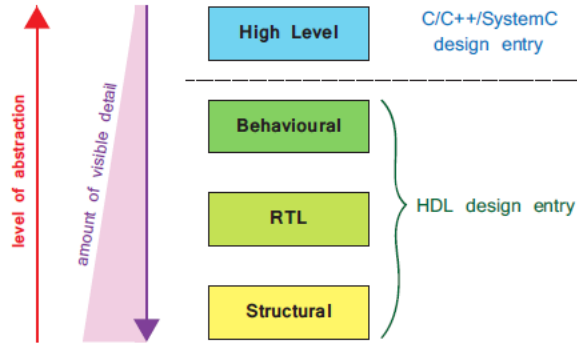
Ενότητα 3

30

- High-level synthesis (HLS)

Επίπεδα αφάιρησης στην σχεδίαση

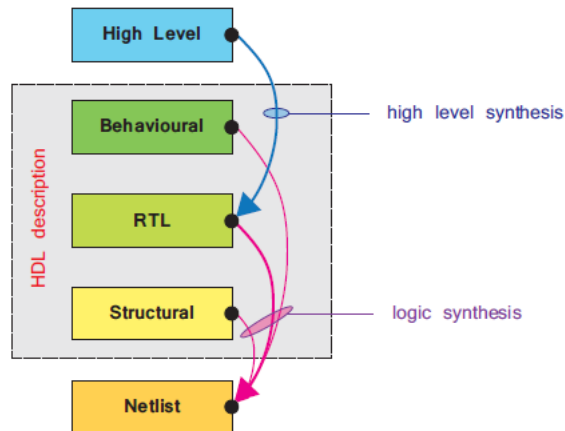
31



Πηγή: The Zynq Book

High-level vs. logic synthesis

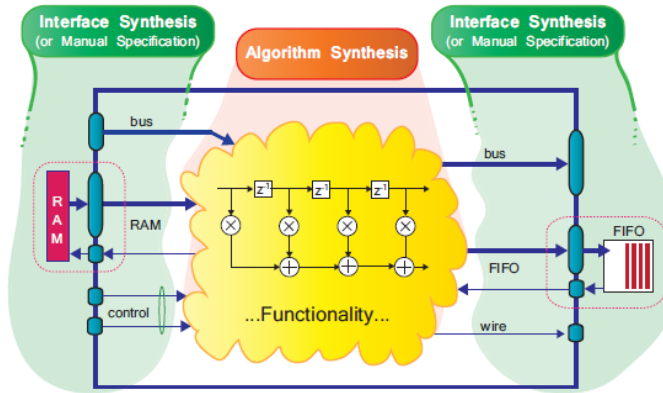
32



Πηγή: The Zynq Book

Αλγόριθμος και διασύνδεση

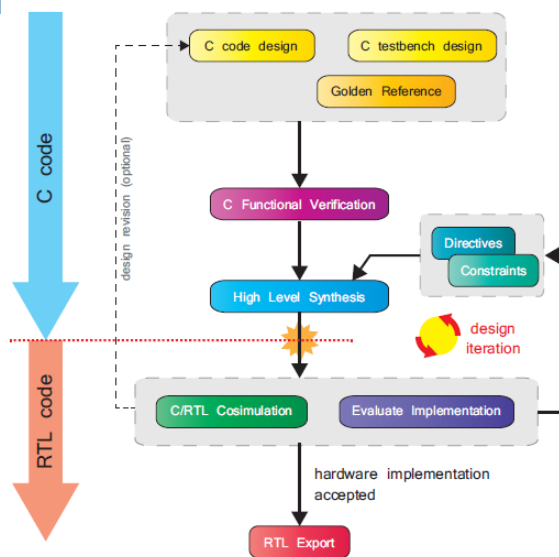
33



Πηγή: The Zynq Book

Ροή σχεδίασης HLS

34



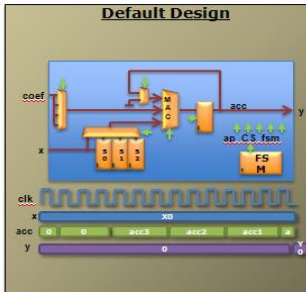
Πηγή: The Zynq Book

Διαφορετικές σχεδιάσεις

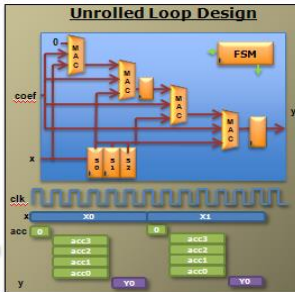
35

```
...  
loop: for (i=3;i>=0;i--) {  
  if (i==0) {  
    acc+=x*c[0];  
    shift_reg[0]=x;  
  } else {  
    shift_reg[i]=shift_reg[i-1];  
    acc+=shift_reg[i]*c[i];  
  }  
}  
....
```

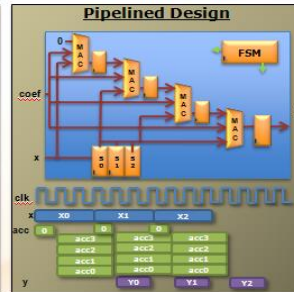
The same hardware is used for each iteration of the loop:
•Small area
•Long latency
•Low throughput



Different hardware is used for each iteration of the loop:
•Higher area
•Short latency
•Better throughput



Different iterations are executed concurrently:
•Higher area
•Short latency
•Best throughput



Πηγή: Xilinx workshop material

HLS: βασική ιδέα

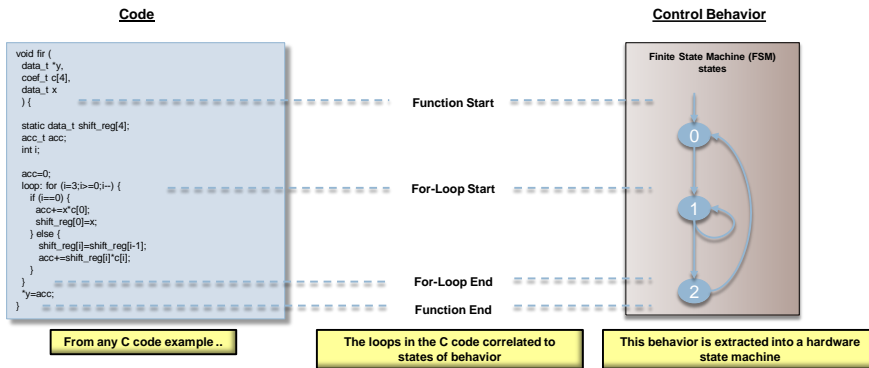
36

- Πώς εξαγεται υλικό από κώδικα C?
 - ▣ Παράγεται η μονάδα ελέγχου (control unit) και η διαδρομή δεδομένων (datapath) σε υψηλό επίπεδο
 - Το ίδιο επαναλαμβάνεται και στις υπομονάδες (π.χ. MAC)
- Πώς η μονάδα ελέγχου και η διαδρομή δεδομένων μετατρέπονται σε κύκλωμα?
 - ▣ Μέσω των διαδικασιών δρομολόγησης (scheduling) και δέσμευσης (binding)

Πηγή: Xilinx workshop material

HLS: Έλεγχος

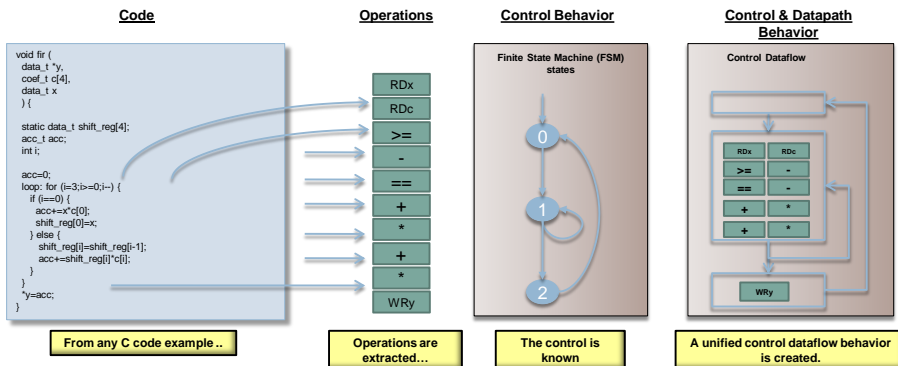
37



Πηγή: Xilinx workshop material

HLS: Έλεγχος και διαδρομή δεδομένων

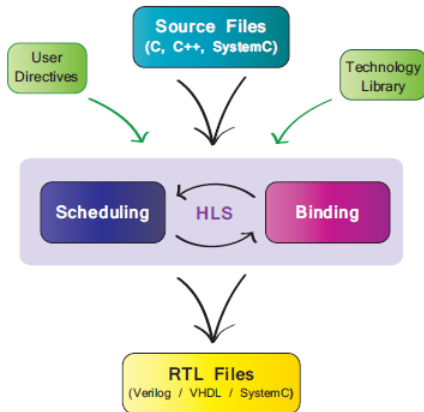
38



Πηγή: Xilinx workshop material

Scheduling και binding

39



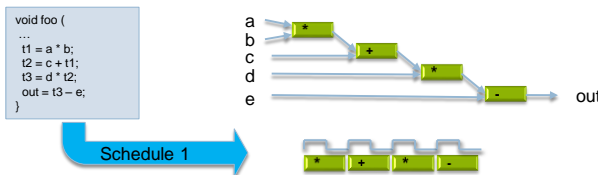
- **Scheduling (δρομολόγηση):**
 - Μετάφραση των εντολών RTL όπως έχουν ερμηνευτεί από τον κώδικα C σε ένα σύνολο λειτουργιών και απόφαση πότε θα εκτελεστούν και πόσο θα διαρκέσουν. Οι αποφάσεις σε αυτό το στάδιο επηρεάζονται από την συχνότητα του ρολογιού, την τεχνολογία της συσκευής και τις οδηγίες/περιορισμούς του χρήστη
- **Binding (δέσμευση):**
 - Συσχέτιση των δρομολογημένων λειτουργιών με φυσικές μονάδες της συσκευής. Τα χαρακτηριστικά (λειτουργικά & χρονικά) αυτών των μονάδων μπορεί να επηρεάσουν την δρομολόγηση και επομένως να χρειαστεί να ξανατρέξει η διεργασία της δρομολόγησης

Πηγή: The Zynq Book

Scheduling

40

- Οι λειτουργίες στον γράφο ροής δεδομένων αντιστοιχίζονται σε κύκλους ρολογιού



- Η τεχνολογία και οι περιορισμοί του χρήστη επηρεάζουν την δρομολόγηση
 - Μια γρηγορότερη τεχνολογία (και ίσως με ένα πιο αργό ρολόι) μπορεί να επιτρέψει να εκτελεστούν περισσότερες λειτουργίες στον ίδιο κύκλο



- Και ο κώδικας μπορεί να επηρεάσει την δρομολόγηση
 - Π.χ. οι εξαρτήσεις δεδομένων πρέπει να ληφθούν υπόψη

Πηγή: Xilinx workshop material

Binding

41

- Οι λειτουργίες (τελεστές) αντιστοιχίζονται σε μονάδες (functional cores) από μια βιβλιοθήκη
- Απόφαση: να μοιραστούν τους πόρους
 - ▣ Με βάση αυτήν την δρομολόγηση:
- Απόφαση: να μην μοιραστούν τους πόρους
 - ▣ Με βάση αυτήν την δρομολόγηση:



- Πρέπει να χρησιμοποιήσει 2 πολλαπλασιαστές, αφού χρησιμοποιούνται στον ίδιο κύκλο
- Μπορεί να αποφασίσει να χρησιμοποιήσει έναν αθροιστή και έναν αφαιρέτη ή έναν κοινό αθροιστή/αφαιρέτη



- Μπορεί να αποφασίσει να μοιραστεί τους πολλαπλασιαστές (χρησιμοποιούνται σε διαφορετικούς κύκλους)
- Ή μπορεί να αποφασίσει ότι το κόστος της κοινής χρήσης (έλεγχος+πολυπλέκτες) μπορεί να επηρεάσουν την απόδοση και δεν συμφέρει

Πηγή: Xilinx workshop material