



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΓΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

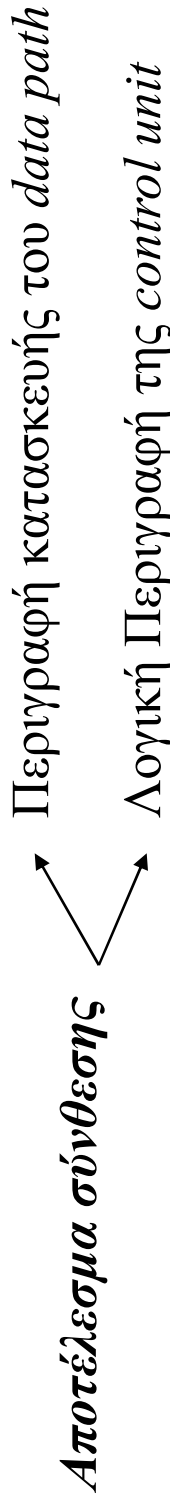


Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

Σύνθεση σε επίπεδο Αρχιτεκτονικής

Εισαγωγή

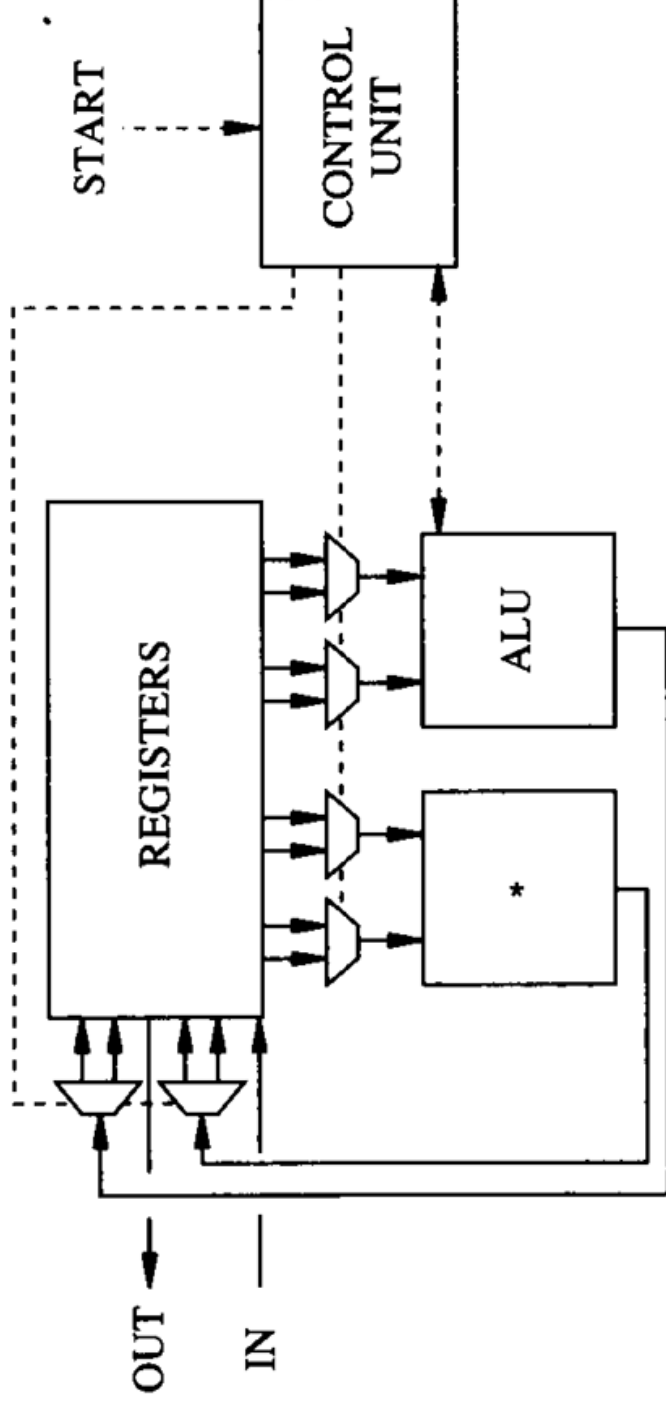
Αρχιτεκτονική Σύνθεση: κατασκευή της μακροσκοπικής δομής ενός ψηφιακού κυκλώματος από μοντέλα συμπεριφοράς (ακολουθιακοί γράφοι – γράφοι ροής δεδομένων).



Data path: διασύνδεση resources (αριθμητικά, λογικά), κυκλωμάτων καθοδήγησης (πολυπλέκτες, δίαυλοι), και καταχωρητών ή μονάδων μνημών.

Control Unit: Μονάδα ελέγχου της μονάδας Data Path.

Εισαγωγή



Εισαγωγή

Βασικά Χαρακτηριστικά Αρχιτεκτονικής Σύνθεσης:

- ✓ Οι υλοποιήσεις εκτιμούνται με:
 - (α) την επιφάνεια (area),
 - (β) την περίοδο ρολογιού (cycle time),
 - (γ) τον αριθμό κύκλων διεκπεραίωσης όλων των λειτουργιών (latency),
 - (δ) τον ρυθμό παραγωγής αποτελεσμάτων (throughput) για pipelined κυκλώματα.
- ✓ Όρια χειρότερης περίπτωσης σε επιφάνεια και απόδοση μπορούν να αποκλείσουν μη επιθυμητές υλοποιήσεις.
- ✓ Μπορούν να χρησιμοποιηθούν προεπιλεγμένες δομές για διάφορες λειτουργίες (αριθμητικές κλπ) με συγκεκριμένα χαρακτηριστικά.

Εισαγωγή

Βασικά Χαρακτηριστικά Αρχιτεκτονικής Σύνθεσης:

- ✓ Η επιφάνεια και η ταχύτητα μπορούν μόνο να εκτιμηθούν προσεγγιστικά σε αυτό το επίπεδο.
- ✓ Εξαρτώνται από τα resources, την λογική καθοδήγησης, τα κυκλώματα αποθήκευσης, την καλωδίωση και τον έλεγχο.
- ✓ *Resource Dominated Circuits*: κυκλώματα στα οποία κάνουμε την υπόθεση ότι η απόδοση εξαρτάται μόνο από τα resources (πχ DSPs).

Οι Προδιαγραφές Σύνθεσης συμπεριλαμβάνουν:

- ✓ μοντέλα συμπεριφοράς (ακολουθιακοί γράφοι),
- ✓ λεπτομέρειες για τα resources
- ✓ περιορισμούς (*constraints*)

Προδιαγραφές για σύνθεση αρχιτεκτονικής

Τα resources κατηγοριοποιούνται ως εξής:

- ✓ *Δειτοργικά (functional)*. Υλοποιούν αριθμητικές και λογικές συναρτήσεις. Διαίρούνται σε δύο υποκατηγορίες:
 - (α) primitive (υποκυκλώματα που χρησιμοποιούνται πολύ συχνά) και
 - (β) application-specific (λύνουν συγκεκριμένα προβλήματα εφαρμογών).
- ✓ *Μνήμης*. Καταχωρητές, μνήμες ανάγνωσης – ανάγνωσης/εγγραφής.
- ✓ *Διασύνδεσης (Interface)*. Δίαυλοι, I/O pads και κυκλώματα διασύνδεσης.

Resource Execution Delay: οι κύκλοι που απαιτούνται από το resource για να εκτελεστεί η λειτουργία του (εξαρτάται από το cycle time).

Resource Dominated κυκλώματα: Απόδοση και επιφάνεια εξαρτώνται από τα resources ενώ steering logic, registers, wiring δεν λαμβάνονται υπόψη

Non-Resource Dominated κυκλώματα: Απόδοση και επιφάνεια εξαρτώνται από τα resources και επίσης τα: steering logic, registers, wiring.

Περιορισμοί (Constraints)

Οι **περιορισμοί** στην σύνθεση αρχιτεκτονικής ανήκουν σε δύο κατηγορίες:

- **Περιορισμοί Διασύνδεσης (Interface):** Καθορίζουν τον τύπο και τον χρονισμό της ανταλλαγής δεδομένων.
- **Περιορισμοί Υλοποίησης:** καθορισμός απιτούμενων ιδιοτήτων του κυκλώματος (πχ επιφάνεια/απόδοση - cycle time/latency).

Resource Binding: μία συγκεκριμένη λειτουργία θα υλοποιηθεί από προεπιλεγμένη resource (προκύπτει από την εμπειρία του σχεδιαστή).

Synthesis from partial structure: Αρχιτεκτονική σύνθεση με περιορισμούς τύπου resource binding.

Στην περίπτωση των resource dominated κυκλωμάτων περιορισμός στον αριθμό των resources = άνω όριο στην επιφάνεια που θα απαιτηθεί.

Προβλήματα αρχιτεκτονικής σύνθεσης

Θεωρούμε ότι ένα κύκλωμα καθορίζεται από:

- Έναν ακολουθιακό γράφο, πολωμένο και άκυκλο με πηγή και καταβόθρα.
- Λειτουργικά resources πλήρως ορισμένα σε επιφάνεια/καθυστέρηση.
- Ένα σύνολο περιορισμών.
- Η αποθήκευση γίνεται με καταχωρητές και οι διασυνδέσεις με καλώδια.

Οι κορυφές του γράφου συμβολίζουν λειτουργίες ενώ οι ακμές συμβολίζουν εξαρτήσεις

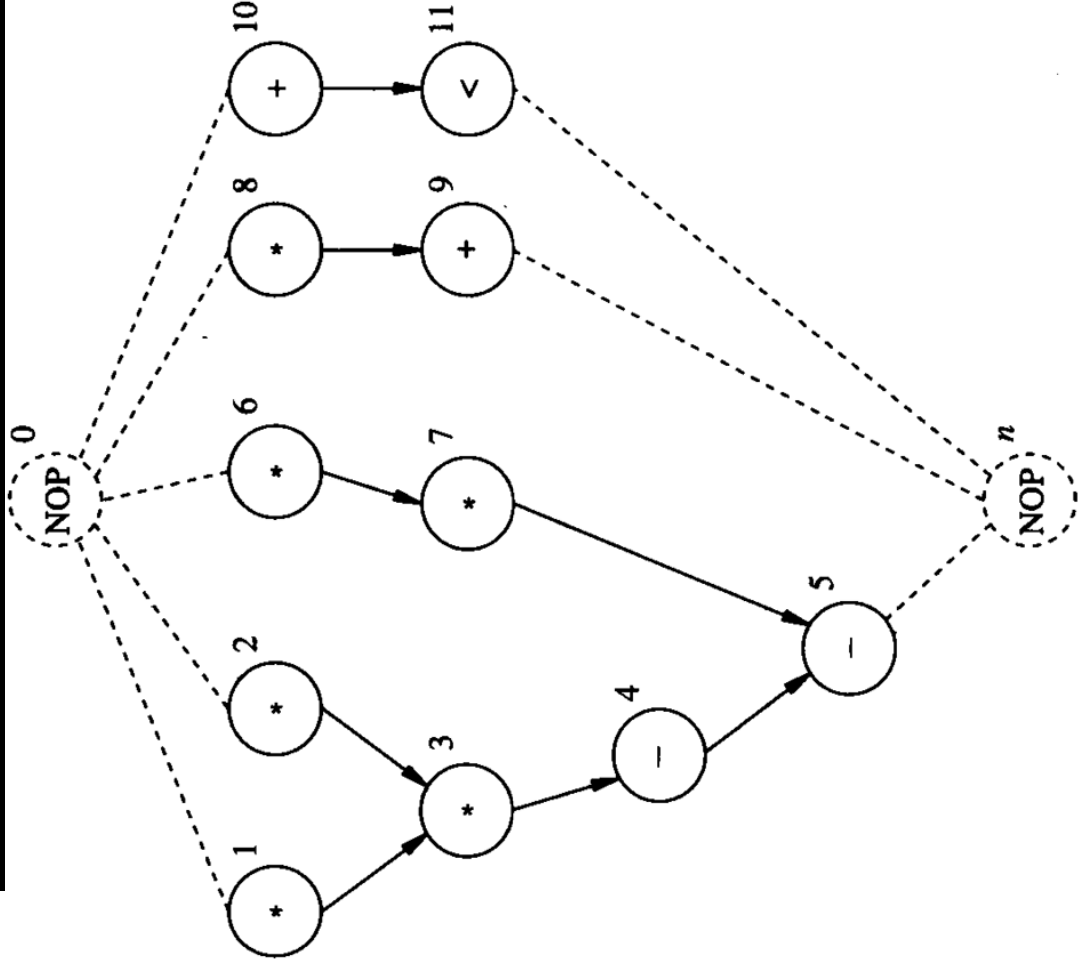
Η σύνθεση αρχιτεκτονικής και η βελτιστοποίηση γίνεται σε δύο στάδια:

1. Τοποθέτηση των λειτουργιών σε τόπο (resources) και χρόνο.
2. Καθορισμός των διασυνδέσεων του data path και των προδιαγραφών λογικού επιπέδου της μονάδας ελέγχου.

Scheduling

- Θεωρούμε ότι :
- Η καθυστέρηση των κορυφών πηγής και καταβόθρας είναι 0.
 - Ο χρόνος εκκίνησης μίας λειτουργίας είναι ο στιγμή που ξεκινά η εκτέλεσή της.
 - **Scheduling**: είναι ο καθορισμός των χρόνων εκκίνησης των λειτουργιών με βάση τις προτεραιότητες του ακολουθιακού γράφου.
 - *Latency*: είναι η διαφορά μεταξύ του χρόνου εκκίνησης της καταβόθρας από τον χρόνο εκκίνησης της πηγής.
 - *Scheduled sequencing graph*: είναι ένας γράφος του οποίου οι κορυφές έχουν ονομαστεί με τον χρόνο εκκίνησης των αντίστοιχων λειτουργιών.
 - Ένα πρόγραμμα (schedule) πρέπει να ικανοποιεί χρονικούς περιορισμούς και περιορισμούς χρήσης των resources.

Scheduling



Θεωρούμε ότι :

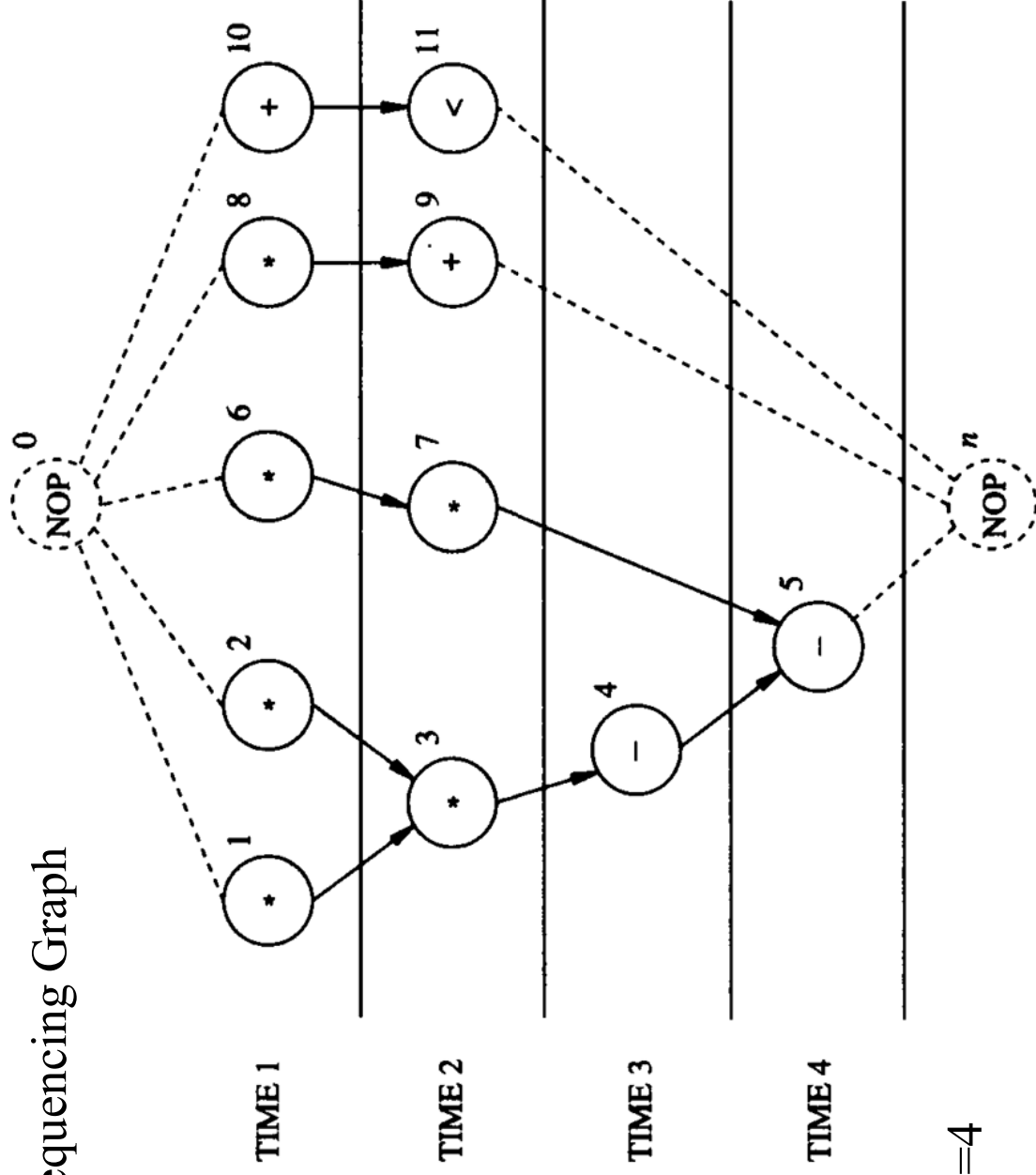
- Όλες οι λειτουργίες έχουν καθυστέρηση μίας χρονικής μονάδας.
- Έστω ότι δεν υπάρχει περιορισμός επιφάνειας (μία μονάδα χρησιμοποιείται ανά λειτουργία).

Λειτουργία Χρόνος Εκκίνησης

$V_1, V_2, V_6, V_8, V_{10}$	1
V_3, V_7, V_9, V_{11}	2
V_4	3
V_5	4

Scheduling

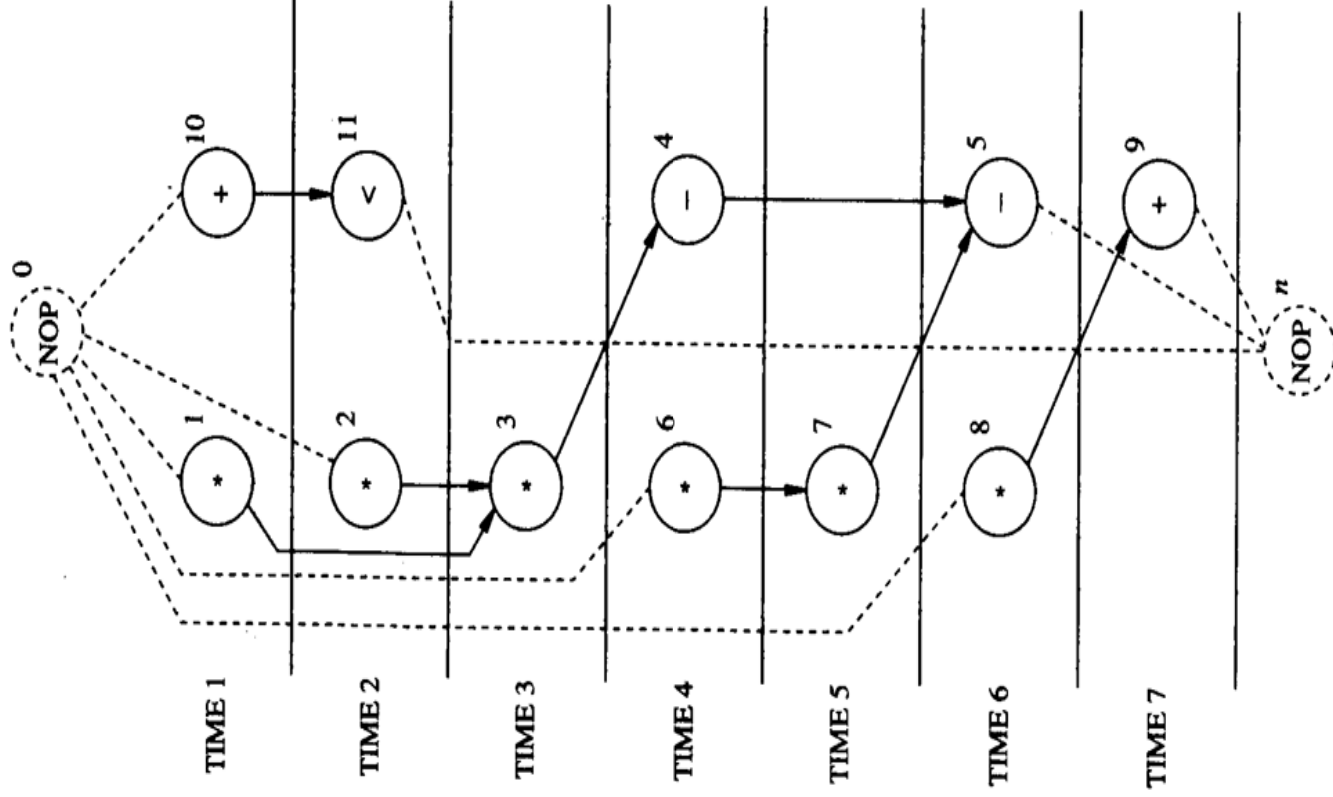
Scheduled Sequencing Graph



$$\text{Latency} = t_n - t_0 = 5 - 1 = 4$$

Scheduling

Έστω ότι υπάρχει περιορισμός επιφάνειας: Scheduled Sequencing Graph με περιορισμούς σε χρήση resources (1 ALU – 1 MUL)



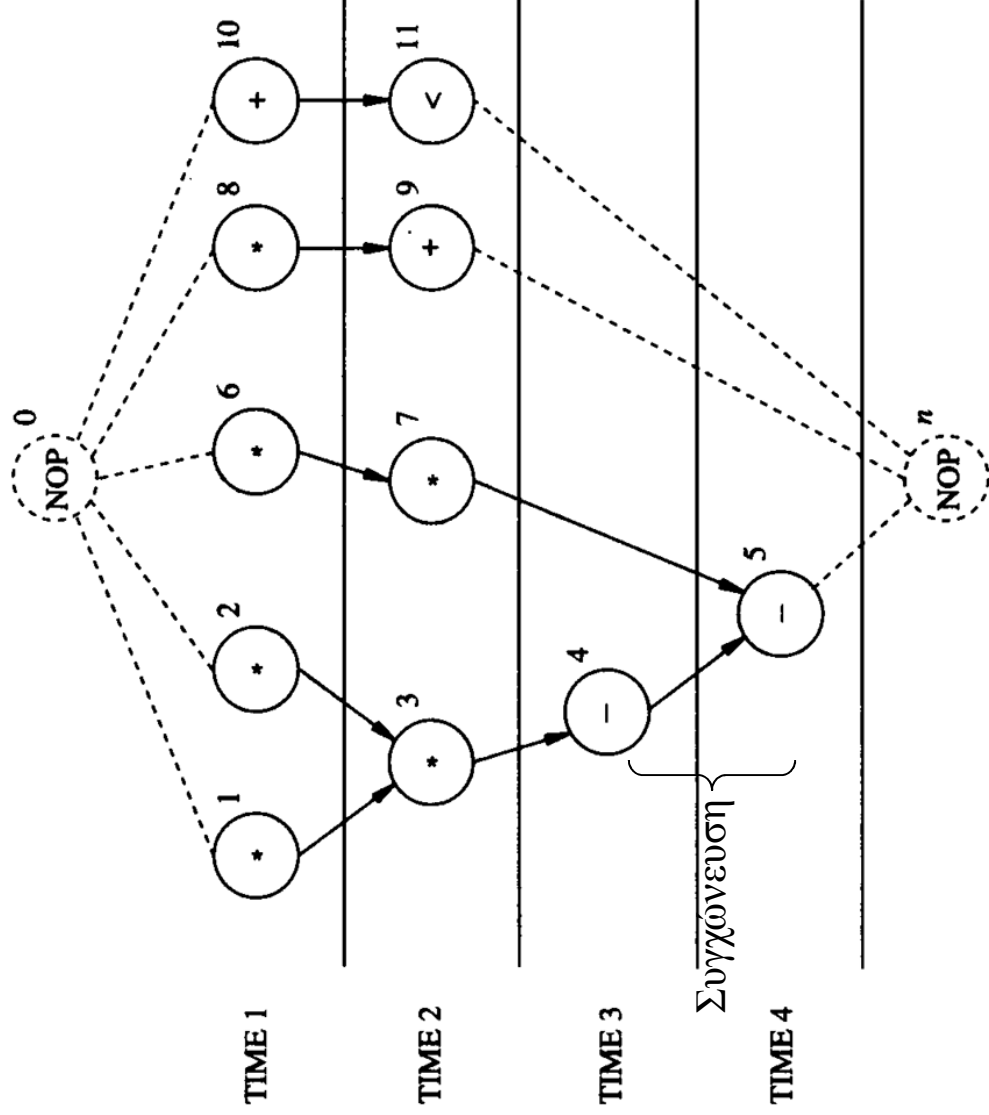
Mul Alu Χρόνος Εκκίνησης

V ₁	V ₁₀	1
V ₂	V ₁₁	2
V ₃	-	3
V ₆	V ₄	4
V ₇	-	5
V ₈	V ₅	6
-	V ₉	7

$$\text{Latency} = t_n - t_0 = 8 - 1 = 7$$

Scheduling

Αντί για ακέραιες μονάδες καθυστέρησης μπορούμε να θεωρήσουμε τους χρόνους διάδοσης των σημάτων μέσα από τα συνδυαστικά resources. Τότε δύο ή περισσότερες λειτουργίες μπορούν να ολοκληρωθούν στον ίδιο κύκλο (chained).



Πχ. για $Mul=35ns$, $Add=25ns$, $cycle\ time=50\ ns$ τότε οι $\{v4, v5\}$ και $\{v10, v11\}$ μπορούν να ολοκληρωθούν σε έναν κύκλο οπότε απαιτούνται 3 κύκλοι

Binding

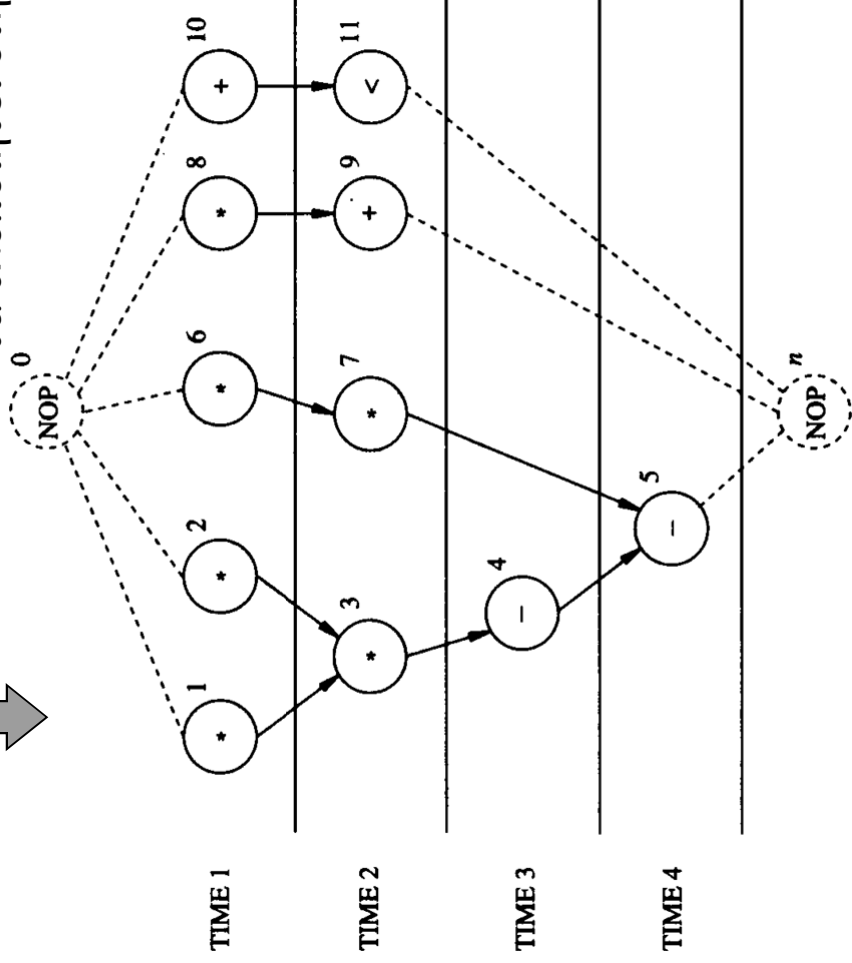
Δέσμευση (Binding): καθορίζει ποιες λειτουργίες θα εκτελεστούν από ποια resources.

- ✓ Είναι αλληλεπένδετο πρόβλημα με το scheduling.
- ✓ Κάποιες φορές γίνεται πριν το scheduling, ενώ άλλες φορές γίνεται μετά.
- ✓ Μπορεί να γίνει και ταυτόχρονα με το scheduling.
- ✓ Η ακριβής υλοποίηση έχει να κάνει με τους περιορισμούς που πρέπει να ικανοποιηθούν (area-delay).

Dedicated Binding

Dedicated Binding. Κάθε λειτουργία ανατίθεται σε διαφορετικό resource

↓ Συμβολισμός: $\beta(v_i)=(t,r)$ η λειτουργία που αντιστοιχεί στην κορυφή v_i θα υλοποιηθεί στην r resource του τύπου t .

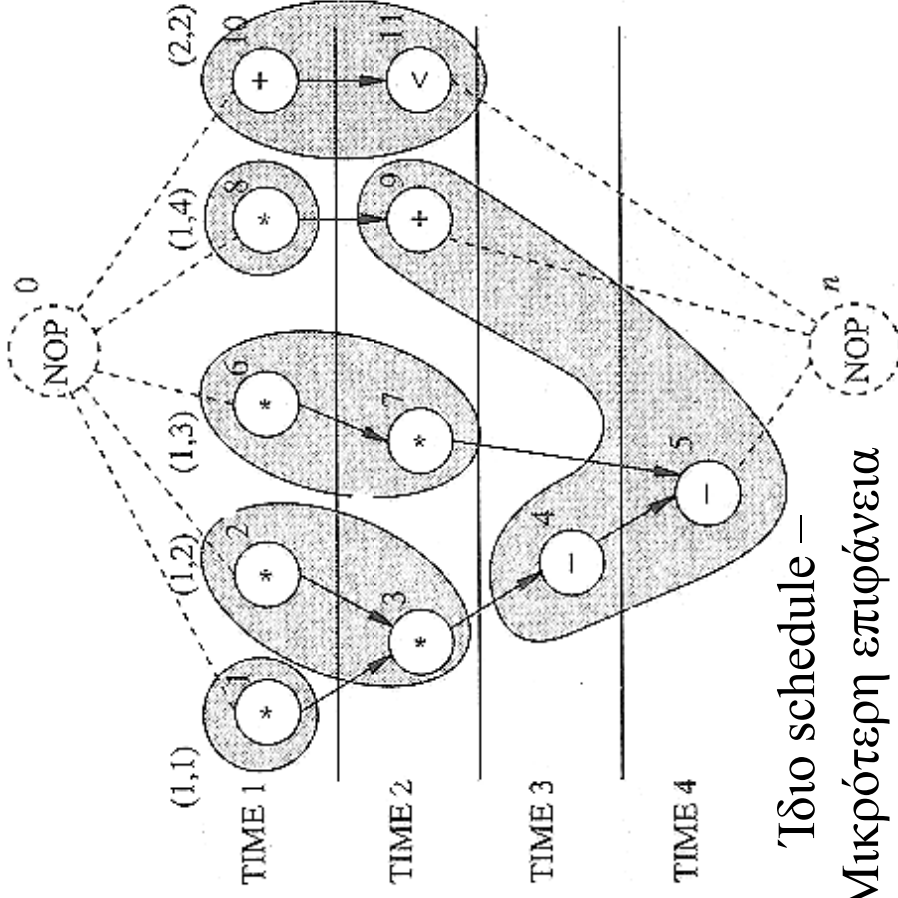


Πχ. Έχουμε 11 λειτουργίες και 11 resources τύπων $\{1,2\} = \{\text{multiplier, ALU}\}$ με την ALU να εκτελεί πρόσθεση, αφαίρεση και συγκρίσεις. Απαιτούνται 6 multipliers και 5 ALUs.

- $\beta(v_1)=(1,1)$ $\beta(v_7)=(1,5)$
- $\beta(v_2)=(1,2)$ $\beta(v_8)=(1,6)$
- $\beta(v_3)=(1,3)$ $\beta(v_9)=(2,3)$
- $\beta(v_4)=(2,1)$ $\beta(v_{10})=(2,4)$
- $\beta(v_5)=(2,2)$ $\beta(v_{11})=(2,5)$
- $\beta(v_6)=(1,4)$

Binding

Για να είναι αποδοτικό το binding τότε πρέπει να διαμοιράζονται οι resources ανάμεσα στις διάφορες λειτουργίες, με τον περιορισμό ότι δύο λειτουργίες που χρησιμοποιούν την ίδια resource δεν εκτελούνται ταυτόχρονα.



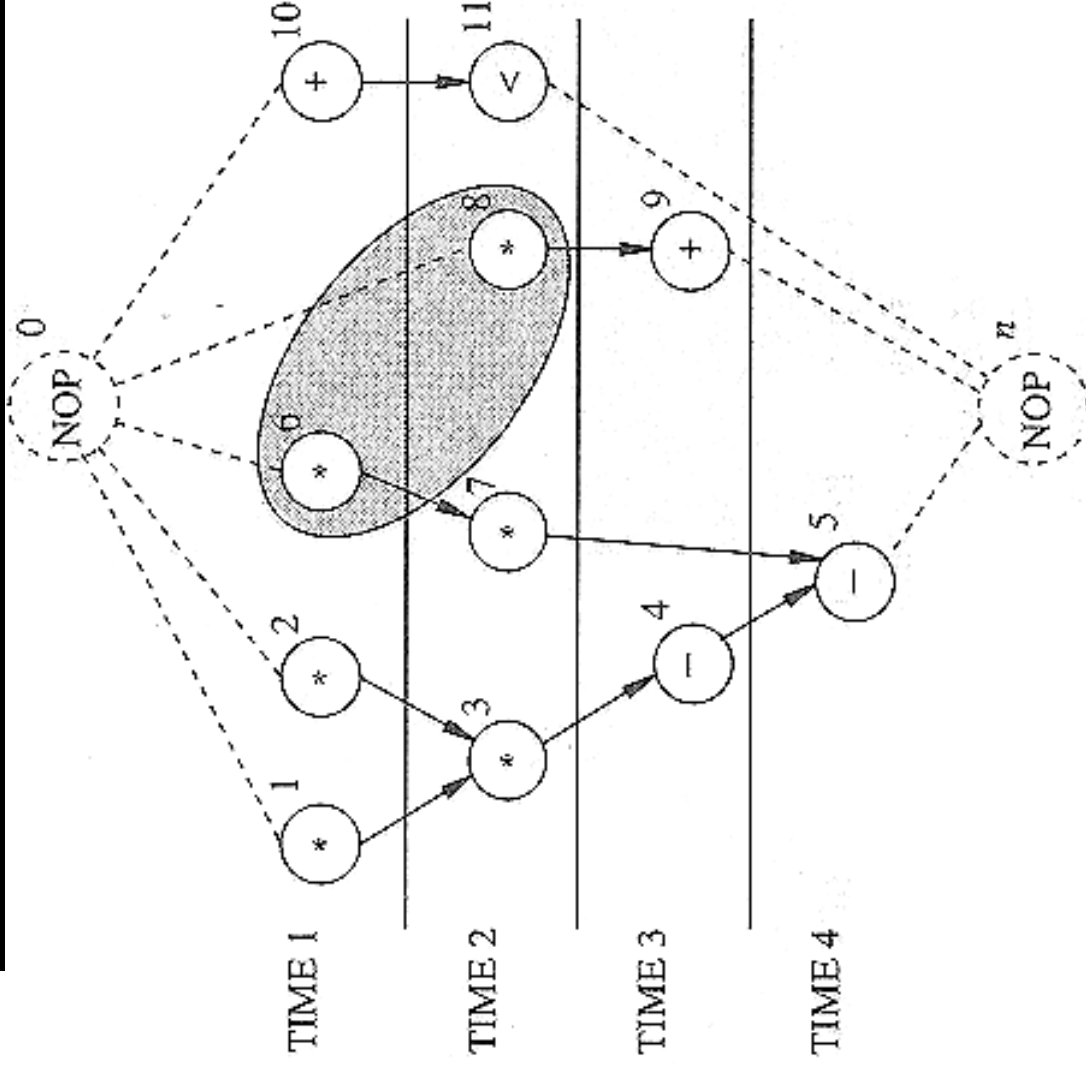
Ίδιο schedule –
Μικρότερη επιφάνεια

Απαιτούνται 4 multipliers και 2 ALUs.

- $\beta(v_1)=(1,1)$ $\beta(v_7)=(1,3)$
- $\beta(v_2)=(1,2)$ $\beta(v_8)=(1,4)$
- $\beta(v_3)=(1,2)$ $\beta(v_9)=(2,1)$
- $\beta(v_4)=(2,1)$ $\beta(v_{10})=(2,2)$
- $\beta(v_5)=(2,1)$ $\beta(v_{11})=(2,2)$
- $\beta(v_6)=(1,3)$

Συνήθης περιορισμοί: άνω όρια στον αριθμό των resources ανά τύπο που μπορούν να χρησιμοποιηθούν

Partial Binding



Κατά το Partial Binding κάποιες λειτουργίες πρέπει να αντιστοιχούνται σε συγκεκριμένα resources (περιορισμοί σχεδιαστή). Τότε το scheduling αλλάζει.

Εάν πρέπει οι λειτουργίες v6 και v8 να εκτελεστούν από τον ίδιο multiplier τότε το scheduling και το partial binding αλλάζουν

Ιεραρχικά Μοντέλα

Scheduling: Ο προγραμματισμός σε ιεραρχικά μοντέλα μπορεί να οριστεί με την ανάθεση χρόνου εκκίνησης σε κάθε κορυφή των γραφών.

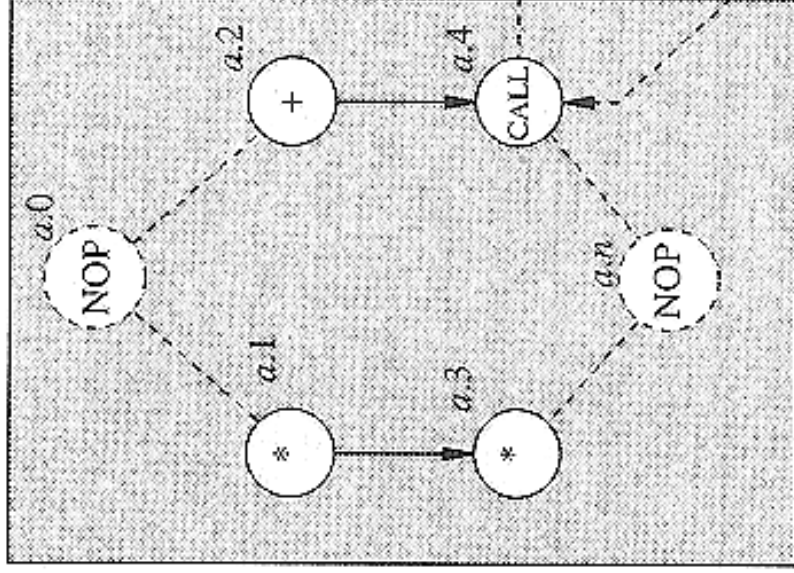
Ο χρόνος εκκίνησης σε μία κορυφή διασύνδεσης (link) είναι ο χρόνος εκκίνησης της κορυφής-πηγής του διασυνδεδεμένου υπογράφου.

Latency: γίνεται με διαπέραση του γραφου από κάτω προς τα πάνω. Στις κορυφές διασύνδεσης έχουμε τις εξής περιπτώσεις:

- **Model call:** latency του υπογράφου
- **Διακλάδωση:** μέγιστη από τις latencies των αντίστοιχων υπογράφων.
- **Επανάληψη:** η latency του σώματος επί τον μέγιστο αριθμό επαναλήψεων.

Binding: είναι το σύνολο όλων των bindings των υπογράφων. Λειτουργίες σε διαφορετικούς υπογράφους μπορούν να διαμοιράζονται resources.

Ιεραρχικά Μοντέλα

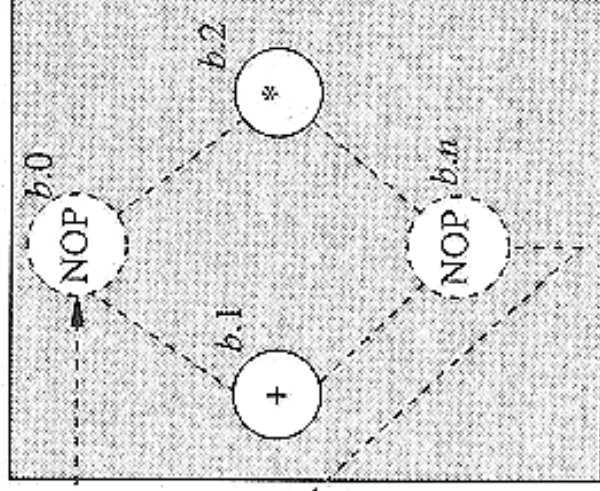


Με resource sharing στην ιεραρχία απαιτούνται 2 Mul και 1 Alu.

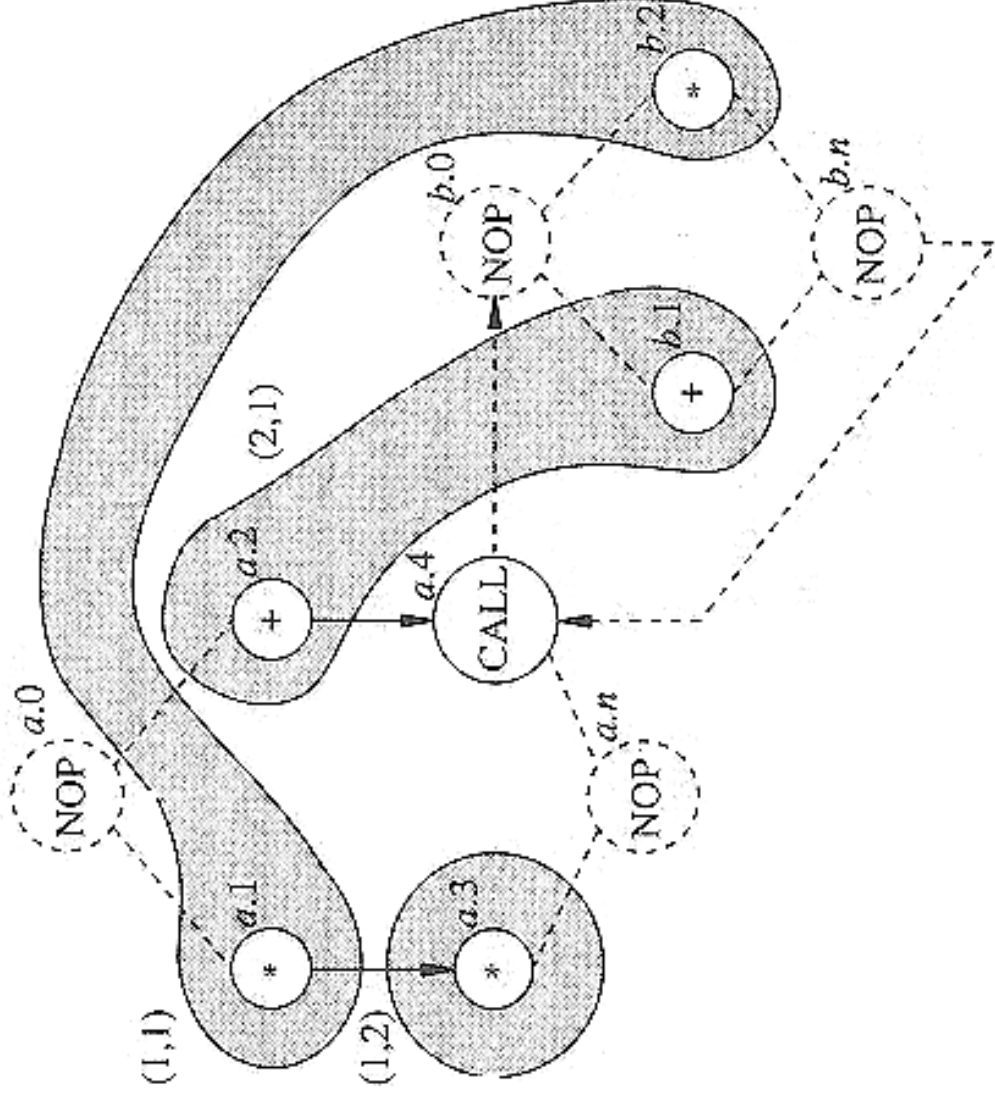
$$\beta(v_{a.1})=(1,1) \quad \beta(v_{b.1})=(2,1)$$

$$\beta(v_{a.2})=(2,1) \quad \beta(v_{b.2})=(1,1)$$

$$\beta(v_{a.3})=(1,2)$$



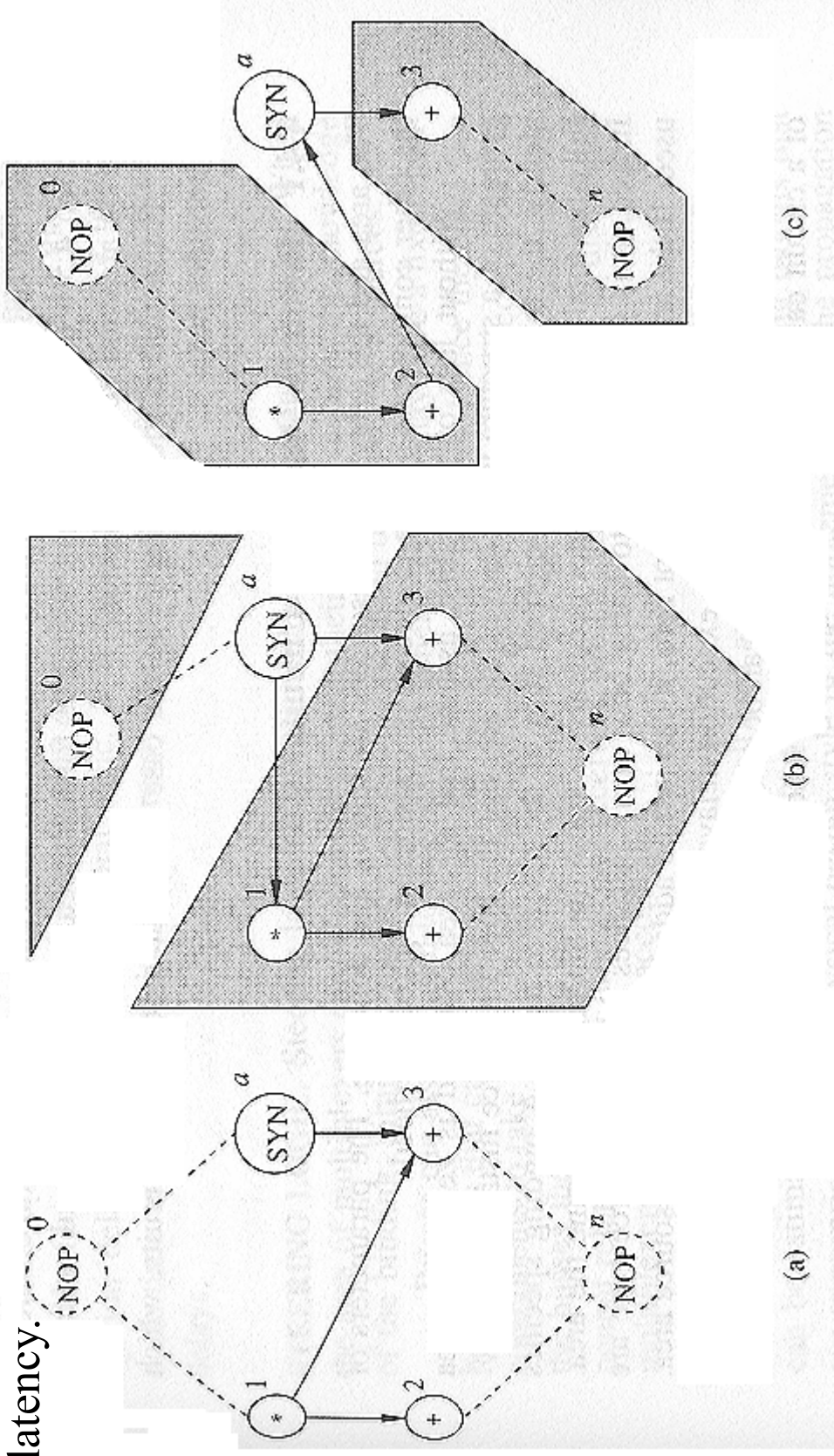
Ιεραρχικά Μοντέλα



Το Πρόβλημα του Συγχρονισμού

Αφορά λειτουργίες που δεν έχουν οριοθετημένη καθυστέρηση όπως εξωτερικός συγχρονισμός και δομές επανάληψης με συνθήκη που εξαρτάται από μεταβαλλόμενα δεδομένα.

Το scheduling σε unbounded γράφους γίνεται με απομόνωση των unbounded λειτουργιών και διαίρεση του γράφου σε υπογράφους που έχουν bounded latency.



Εκτίμηση Επιφάνειας/Απόδοσης

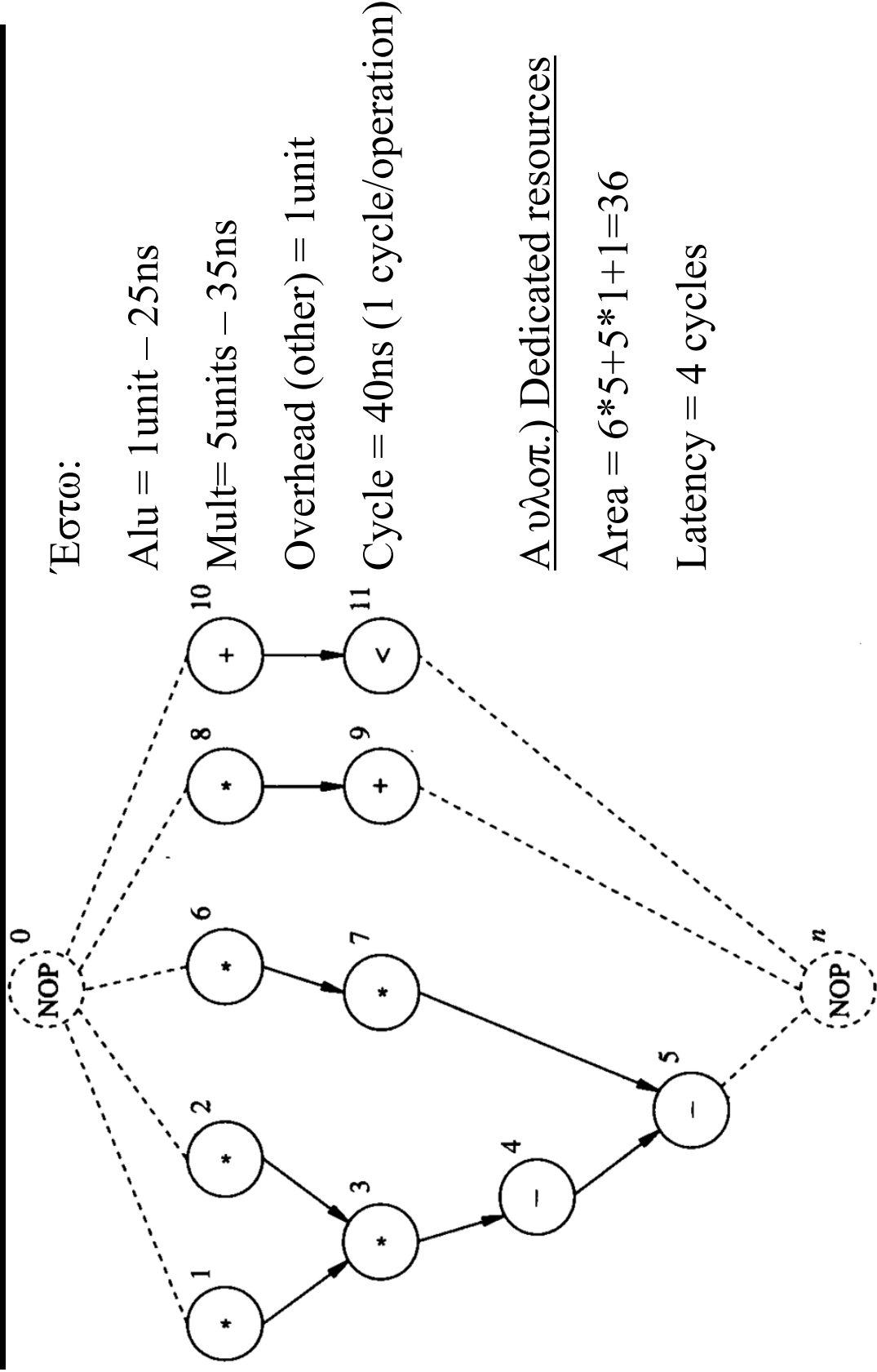
Εκτιμήσεις επιφάνειας/απόδοσης:

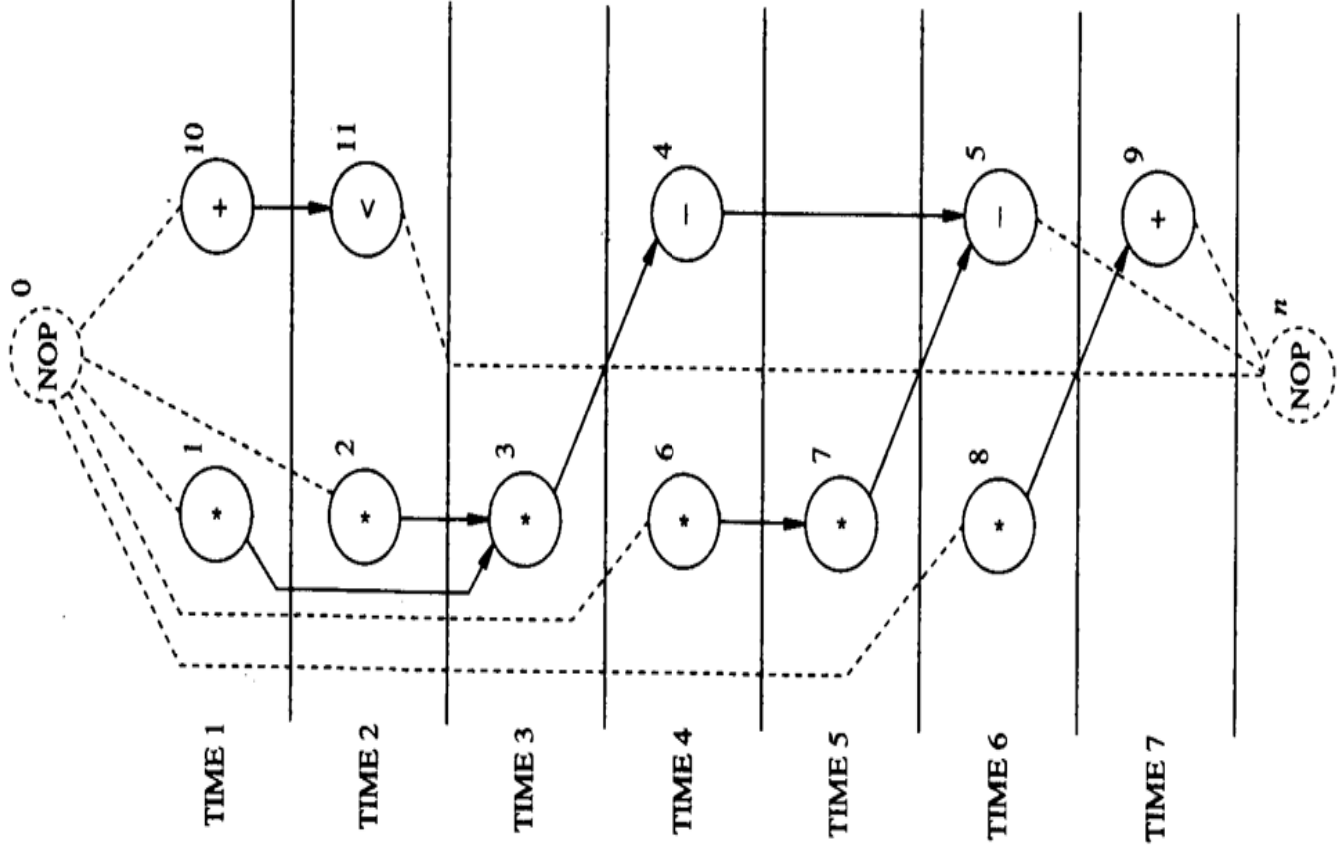
- ✓ Οι εκτιμήσεις σε επιφάνεια και απόδοση είναι απαραίτητες κατά το scheduling και το binding για την λήψη αποφάσεων.
- ✓ Είναι ένα δύσκολο έργο και εξαρτάται από το είδος του κυκλώματος.

Resource Dominated circuits

- ✓ Η επιφάνεια και η απόδοση των resources που κυριαρχούν στον σχεδιασμό είναι γνωστές.
- ✓ Οι παράμετροι των υπόλοιπων components μπορούν να αγνοηθούν ή να θεωρηθούν σταθερές.
- ✓ Για τον υπολογισμό της επιφάνειας αρκεί να γνωρίζουμε πόσες resources από κάθε τύπο έχει το κύκλωμα.
- ✓ Για τον υπολογισμό του latency απαιτείται το schedule.

Εκτίμηση Επιφάνειας/Απόδοσης





Έστω:

Alu = 1 unit – 25ns

Mult= 5units – 35ns

Overhead (other) = 1 unit

Cycle = 40ns (1 cycle/operation)

Βολοπ.) Single resource

Area = 5+1+1=7units

Latency = 7 cycles



Εκτίμηση Επιφάνειας/Απόδοσης

Γενικά Κοκλώματα

✓ Η ολική επιφάνεια είναι το άθροισμα των :

- (α) bounded resources,
- (β) των κυκλωμάτων καθοδήγησης,
- (γ) των μονάδων αποθήκευσης,
- (δ) της μονάδας ελέγχου και
- (ε) των διασυνδέσεων.

✓ Η ολική latency εξαρτάται από το schedule και επιπλέον από τις καθυστερήσεις διάδοσης των μονοπατιών μεταξύ καταχωρητών.

Εκτίμηση Επιφάνειας/Απόδοσης

Καταχωρητές

- Αποθηκεύουν τα δεδομένα που θα πρέπει να μεταφερθούν από μία resource σε μία άλλη μεταξύ δύο διαδοχικών κύκλων.
- Μπορούν να διαμοιραστούν σε διάφορα resources.
- Εξαρτώνται από το scheduling και το binding.
- Επηρεάζουν τον κύκλο ρολογιού αφού προσαυξάνεται με τους χρόνους set-up και propagation delay.

Εκτίμηση Επιφάνειας/Απόδοσης

Λογική Καθοδήγησης (Steering Logic)

- Επηρεάζει και την επιφάνεια και την ταχύτητα.
- Η επιφάνεια των πολυπλεκτών είναι γνωστή. Ο αριθμός τους εξαρτάται από το binding.
- Οι πολυπλέκτες προσαρξάνουν την καθυστέρηση του κυκλώματος.
- Οι Δίαυλοι χρησιμοποιούνται για καθοδήγηση δεδομένων. Λαμβάνονται υπόψη μαζί με τους οδηγούς τους.

Διασυνδέσεις

- Η επιφάνεια των διασυνδέσεων μπορεί να υπολογιστεί από την δομή του κυκλώματος όταν ολοκληρωθεί το binding με χρήση **στατιστικών μοντέλων** (δεν είναι γνωστή η φυσική υλοποίηση).
- Η καθυστέρηση των διασυνδέσεων είναι υπολογίσιμη.

Εκτίμηση Επιφάνειας/Απόδοσης

Μονάδα Ελέγχου

- Επιβαρύνει αρκετά την συνολική επιφάνεια και την καθυστέρηση.
- Υπάρχουν μοντέλα που εκτιμούν την επιφάνεια/απόδοση της μονάδας ελέγχου ανάλογα με την υλοποίηση της:
 - α) με μνήμη μόνο ανάγνωσης (μικροπρογραμματιζόμενη) και
 - β) καλωδιωμένη.
- Πχ η καλωδιωμένη υλοποίηση μοντελοποιείται με μηχανές πεπερασμένου αριθμού καταστάσεων (Παράδειγμα: η εργασία σας).
- Τα μοντέλα εκτίμησης της μονάδας ελέγχου παρέχουν μη-ακριβή αποτελέσματα επειδή οι βελτιστοποιήσεις μπορεί να αλλάξουν πολύ το τελικό αποτέλεσμα

Βελτιστοποίηση Αρχιτεκτονικής

Στόχος Βελτιστοποίησης: είναι ο καθορισμός ενός schedule και ενός binding που θα ικανοποιούν τους περιορισμούς και θα βελτιστοποιούν τις παραμέτρους (area, latency, cycle-time).

Μερική βελτιστοποίηση: Όταν είναι καθορισμένο το schedule ή το binding και βελτιστοποιούμε το άλλο.

Διερεύνηση των λύσεων

- Για διάφορες τιμές του cycle-time (Constraint: Input to register, register to register, register to output) μελετάμε το area/latency trade-off με λύση διάφορων προβλημάτων scheduling.
- Για διάφορα bindings μελετάμε το cycle-time/latency trade-off.
- Για διάφορα schedules μελετάμε το area/cycle-time.

Βελτιστοποίηση Area/Latency

Πρόβλημα: για δοσμένο cycle-time μελετάμε τις διάφορες αποδόσεις σε σχέση με την διακύμανση των παραμέτρων area/latency.

Resource Dominated circuits

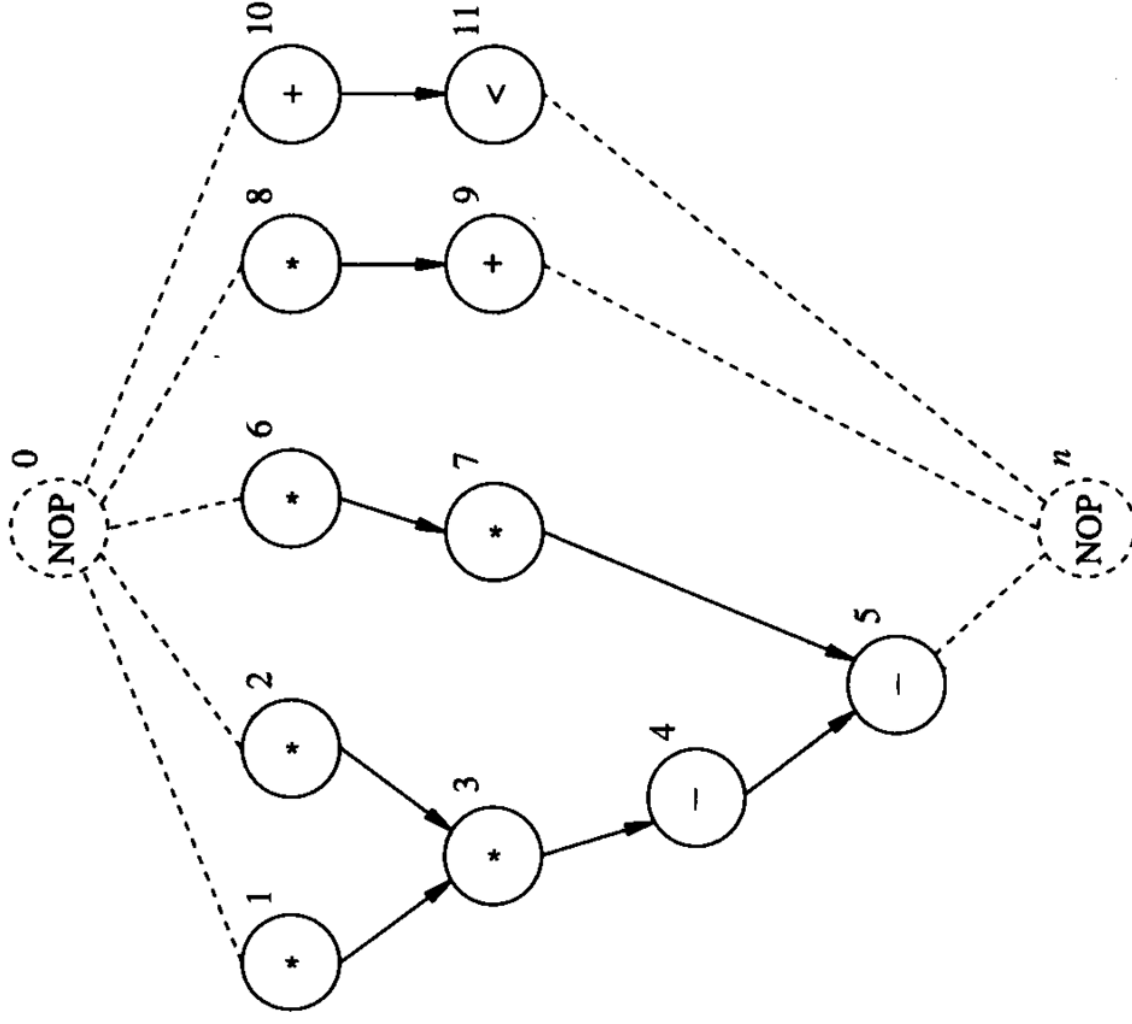
- ✓ Η επιφάνεια εδώ εξαρτάται αποκλειστικά από τα resources που χρησιμοποιούνται.
- ✓ Τα Pareto Points είναι τα ζεύγη (area, latency).
- ✓ Οι ακραίες λύσεις αυτές που αντιστοιχούν στο scheduling για minimum latency και για minimum resource.
- ✓ Ενδιάμεσες λύσεις μπορούν να βρεθούν με λύση των προβλημάτων scheduling:
 - a) resource constrained/minimum latency,
 - b) latency constrained/minimum resource

Βελτιστοποίηση Area/Latency

Γενικά Κυκλώματα

- ✓ Υποθέτουμε ότι η λογική ελέγχου και οι διασυνδέσεις δεν επηρεάζουν ταχύτητα/επιφάνεια.
- ✓ Λαμβάνουμε υπόψη μόνο resources, καταχωρητές και λογική καθοδήγησης.
- ✓ Η δυσκολία του προβλήματος αυξάνει κατακόρυφα όταν συνυπολογίζεται και η λογική ελέγχου καθώς και η διασύνδεση γιατί απαιτείται φυσική υλοποίηση του κυκλώματος.
- ✓ Τα συστήματα CAD είτε εκτελούν scheduling πριν το binding είτε ανάποδα για απλοποίηση του αλγορίθμου.
- ✓ Η εκτέλεση του binding πριν το scheduling επιτρέπει την εκτίμηση της λογικής καθοδήγησης και την ακριβέστερη εκτίμηση των καθυστερήσεων.

Βελτιστοποίηση Area/Latency



Πχ. Έστω $Alu = 1 \text{ unit} - 25 \text{ ns}$,
 $Mult = 5 \text{ units} - 35 \text{ ns}$,

$Overhead (other) = 1 \text{ unit}$

Περιορισμοί:
Επιφάνεια $< 20 \text{ units}$,
 $latency < 8 \text{ cycles}$

Για $cycle \text{ time} = 40 \text{ ns}$ όλες οι
 $resources$ έχουν 1 cycle
καθυστέρηση.

Βελτιστοποίηση Area/Latency

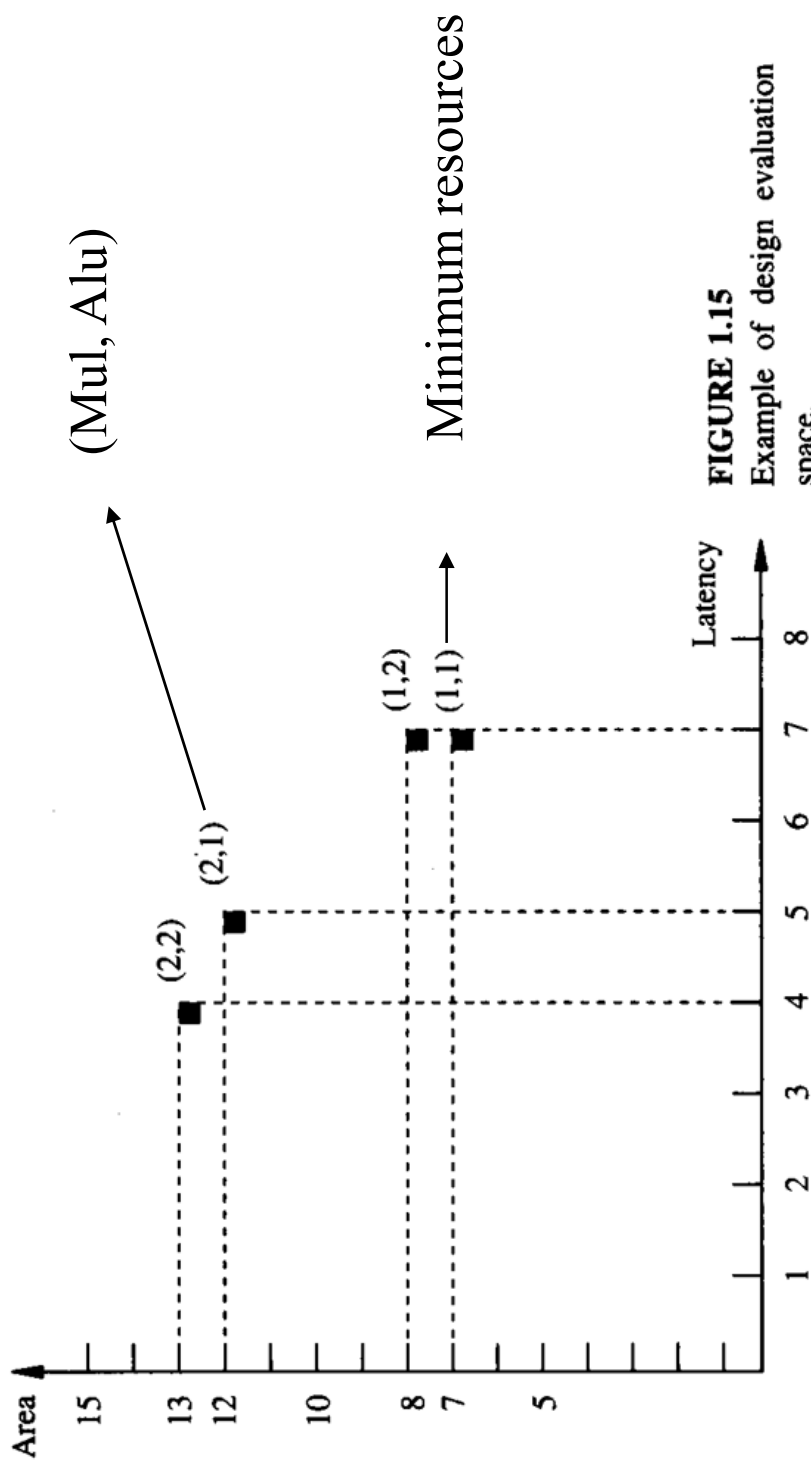
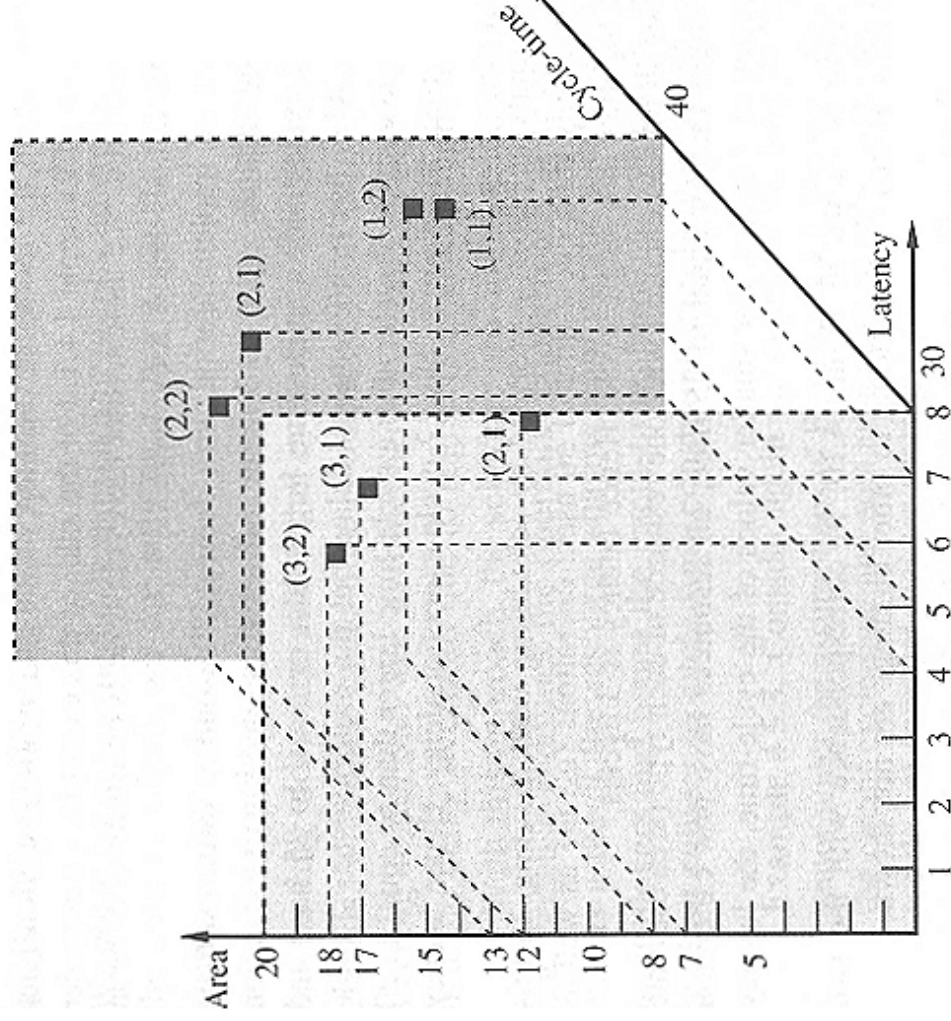


FIGURE 1.15
Example of design evaluation space.

Cycle-time=40ns

Βελτιστοποίηση Area/Latency



Για Cycle-time=30ns, έχουμε
Mult=2cycles, Add=1cycle και μία
δομή με 1 Mult έχει Latency=13cycles
(παραβίαση περιορισμού)

Μία δομή με 2 Mult και 1 Adder έχει
Latency=8 cycles, Area=12units.

Μία δομή με 3 Mult και 1 Adders έχει
Latency=7 cycles, Area=17units.

Μία δομή με 3 Mult και 2 Adders έχει
Latency=6 cycles, Area=18units.

Επιπλέον υλικό παραβιάζει τον
περιορισμό επιφάνειας

Βελτιστοποίηση Cycle-Time/Latency

Μελετούμε την περίπτωση που έχει γίνει το binding και έχει δημιουργηθεί ο οριοθετημένος ακολουθιακός γράφος.

Resource Dominated Circuits

- ✓ Για κάθε τιμή του cycle-time υπολογίζουμε τις καθυστερήσεις εκτέλεσης των λειτουργιών.
- ✓ Κατόπιν δημιουργούμε ένα schedule ελάχιστης latency.
- ✓ Μπορούμε να δημιουργήσουμε scheduling με chaining για βελτίωση.
- ✓ Για δεδομένο Latency θέλουμε να βρούμε το ελάχιστο δυνατό cycle time.
- ✓ Το cycle time υπολογίζεται από το μέγιστο μονοπάτι μεταξύ καταχωρητών.

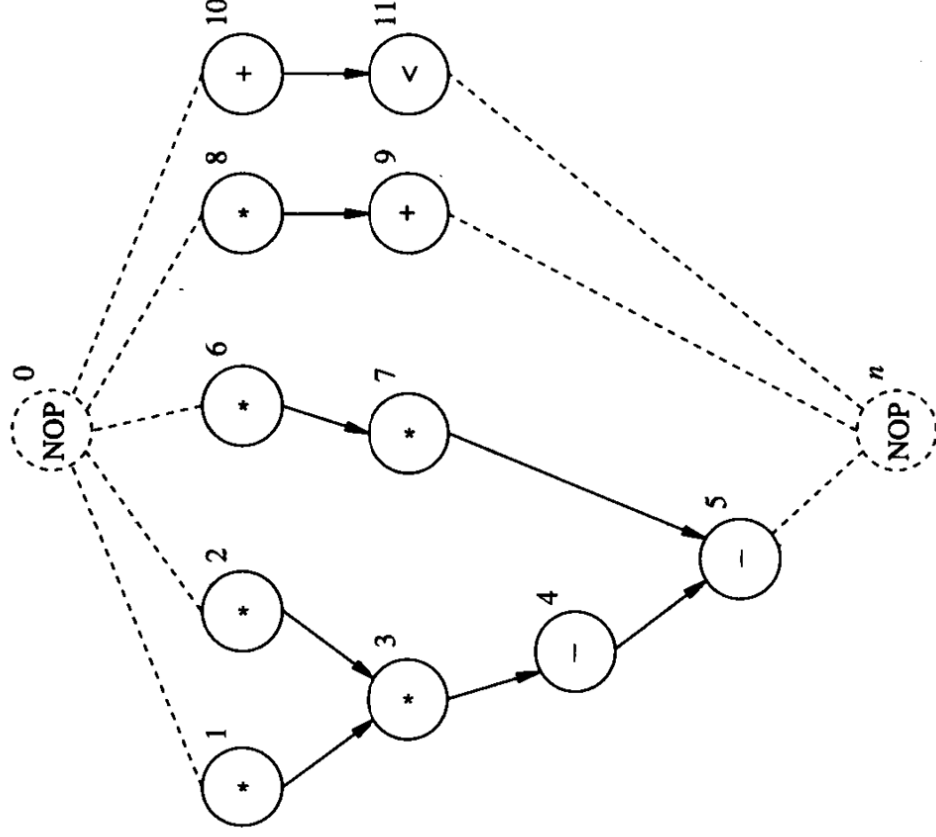
Πχ. Έστω Alu = 1 unit – 25ns, Mult= 5units – 35ns, Overhead (other) = 1 unit

Περιορισμοί: Cycle Time=20-50ns, latency<8cycles

Υπόθεση A: υλοποίηση με dedicated resources (Επιφάνεια=36units).

Βελτιστοποίηση Cycle-Time/Latency

Dedicated Resources



Για $\text{Cycle-time} \geq 35\text{ns}$, έχουμε
 $\text{Mult}=1\text{ cycle}$, $\text{Add}=1\text{ cycle}$ οπότε
 $\text{Latency}=4\text{ cycles}$

Για $35 > \text{Cycle-time} \geq 25\text{ns}$, έχουμε
 $\text{Mult}=2\text{ cycles}$, $\text{Add}=1\text{ cycle}$ οπότε
 $\text{Latency}=6\text{ cycles}$

Για $25 > \text{Cycle-time} \geq 20\text{ns}$, έχουμε
 $\text{Mult}=2\text{ cycles}$, $\text{Add}=2\text{ cycles}$ οπότε
 $\text{Latency}=8\text{ cycles}$

Με chaining: θεωρούμε $\text{cycle time} = 50\text{ns}$ οπότε οι $\{v_4, v_5\}, \{v_{10}, v_{11}\}$ μπορούν να εκτελεστούν σε ένα κύκλο οπότε $\text{Latency}=3\text{ cycles}$.

Βελτιστοποίηση Area/Cycle-Time

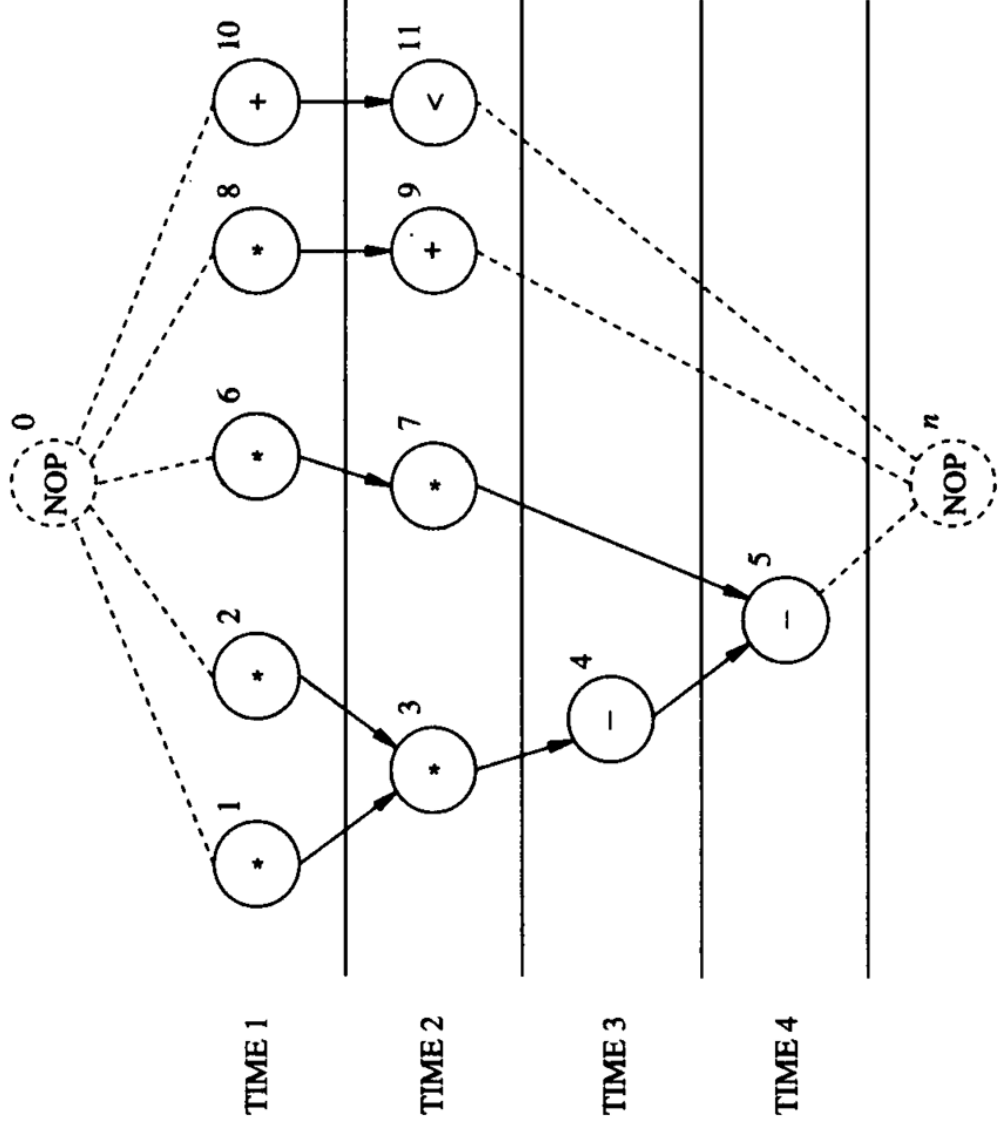
Μελετούμε την περίπτωση που το Latency είναι **προκαθορισμένο**.
Μεταβάλλουμε area και cycle time για την εκτίμηση των λύσεων.

Γενικά κυκλώματα

- ✓ Το κάτω όριο του cycle time καθορίζεται από τα resources, την λογική καθοδήγησης, καταχωρητές κλπ.
- ✓ Η καθυστέρηση των resources και των καταχωρητών είναι δεδομένη,
- ✓ Μας ενδιαφέρει κυρίως η καθυστέρηση της λογικής καθοδήγησης (πολυπλέκτες κλπ) η οποία μεταβάλλεται ανάλογα με το επιλεγμένο binding.

Πχ. Έστω $Alu = 1 \text{ unit} - 25 \text{ ns}$, $Mult = 5 \text{ units} - 35 \text{ ns}$, $Overhead (other) = 1 \text{ unit}$

Βελτιστοποίηση Area/Cycle-Time



Για απλοποίηση θεωρούμε ότι το Cycle Time και η επιφάνεια εξαρτώνται από resources και λογική καθοδήγησης.

Για την περίπτωση που έχουμε dedicated resources δεν απαιτούνται πολυπλέκτες οπότε $area = 3 \text{ units}$, ελάχιστη τιμή του cycle time = 35ns (καθυστέρηση αργότερης resource).

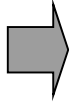
Λύση: $(area, cycle\ time) = (36, 35)$

Βελτιστοποίηση Area/Cycle-Time

Η λύση με dedicated resources είναι πολύ ακριβή σε επιφάνεια.



Προσπαθούμε να μειώσουμε τα resources, χωρίς όμως να αυξήσουμε το latency το οποίο θεωρείται σταθερό.

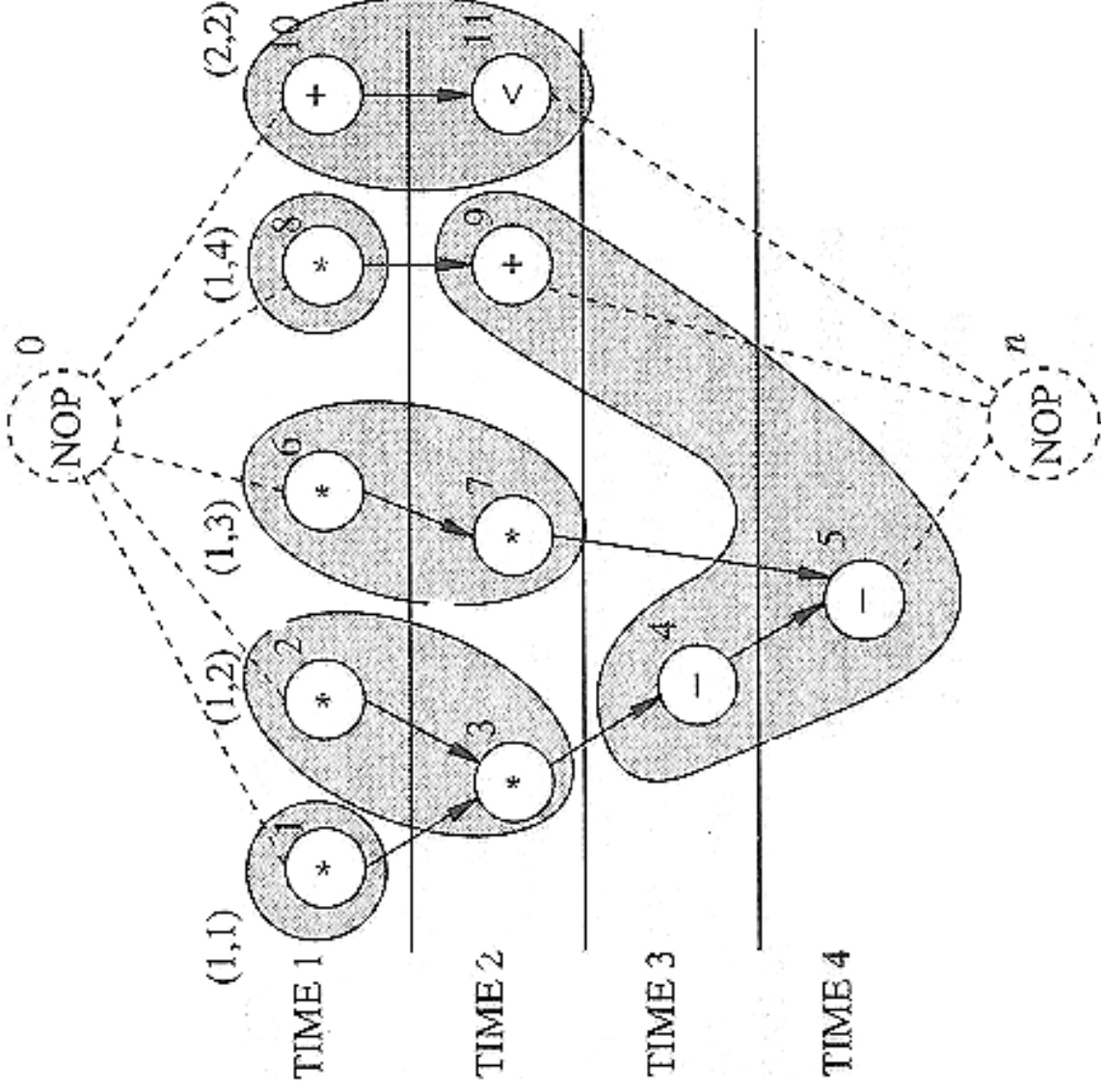


Χρησιμοποιούμε κοινές Resources για λειτουργίες που δεν χρειάζεται να εκτελεστούν παράλληλα.



Κάποιοι πολυπλέκτες πρέπει να μπουν στις εισόδους των resources, και άρα επηρεάζεται το cycle time.

Βελτιστοποίηση Area/Cycle-Time



Για την περίπτωση στο σχήμα θέλουμε 1 πολυπλέκτη τριών εισόδων (0.3units-3ns) και 3 πολυπλέκτες 2 εισόδων (0.2units-2ns).

Συνολική

$$\text{επιφάνεια} = 4 * 5 + 2 * 1 + 1 + 1 * 0.3 + 3 * 0.2 = 23.9$$

Ελάχιστο cycle time = 37ns

(Mul+Mux2) ο πολλαπλα.

καθορίζει το ελάχιστο cycle time

$$\text{Λύση: (area, cycle time) = (23.9, 37)}$$

Βελτιστοποίηση Area/Cycle-Time

Για την περίπτωση στο σχήμα (ένα resource από τον κάθε τύπο) θέλουμε 2 πολυπλέκτες πέντε εισόδων (0.5units-5ns) (ο αριθμός εξαρτάται από τις μεταβλητές εισόδου).

Συνολική επιφάνεια=5+1+1+2*0.5=8units

Συνολική καθυστέρηση=40ns (Mul+Mux5)

Λύση: (area, cycle time) = (8, 40)

