Σχεδιασμός Ολοκληρωμένων Κυκλωμάτων VLSI II

Επιμέλεια:

Βασίλης Παλιουράς, Αναπληρωτής Καθηγητής

Ανδρέας Εμερετλής, Υποψήφιος Διδάκτορας

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Σημείωμα Αδειοδότησης

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη Δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

Ανάπτυξη

Το παρόν εκπαιδευτικό υλικό αναπτύχθηκε στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών του Πανεπιστημίου Πατρών.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

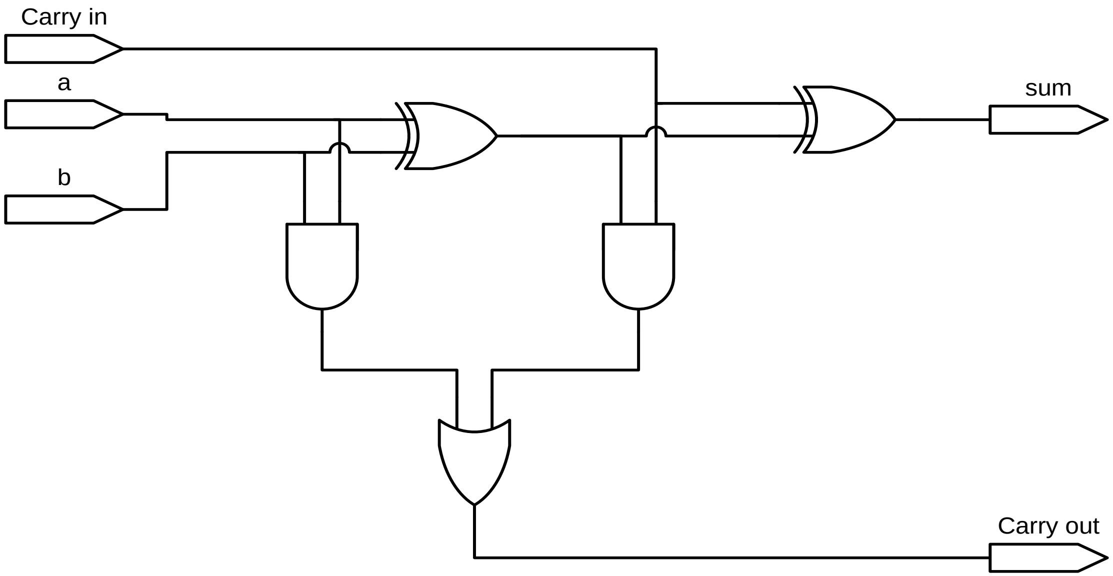
Αρχιτεκτονικές Αθροιστών

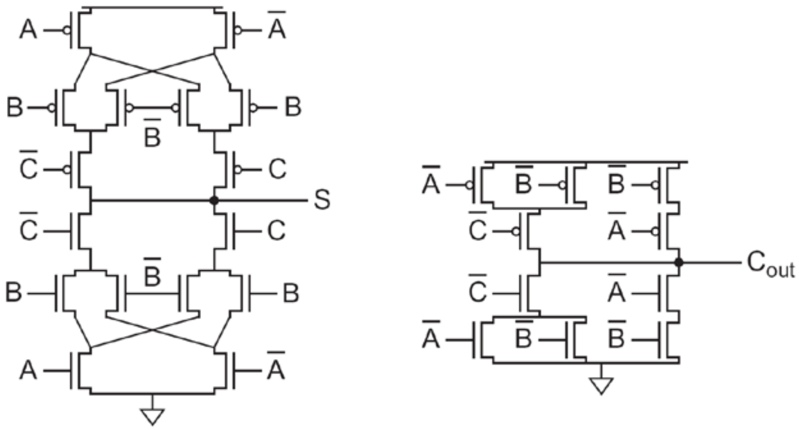
Περίγραμμα Παρουσίασης

- Εισαγωγή
- Αθροιστές διάδοσης κρατουμένου
- Αθροιστές παράλληλου προθέματος
- Άθροιση πολλών εισόδων

Το Σύνηθες Δομικό Στοιχείο

➢ Πλήρης Αθροιστής Ενός Δυαδικού Ψηφίου





Πρόσθεση Αριθμών χωρίς Πρόσημο

$$
\begin{array}{cccccc}
 & \boxed{1} & \boxed{1} & \boxed{1} & & \\
 & & 0 & 1 & 1 & 1 \\
+ & & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 0 & 1
\end{array}
$$

a	b	c_{in}	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Αλγόριθμος Πρόσθεσης στο Επίπεδο του Δυαδικού Ψηφίου

$$S = A + B$$

$$A = \sum_{i=0}^{n-1} a_i 2^i , B = \sum_{i=0}^{n-1} b_i 2^i \quad a_i , b_i \in \{0,1\}$$

$$S = \sum_{i=0}^{n-1} s_i 2^i$$

$$s_i = a_i \oplus b_i \oplus c_i$$

$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

$$i = 0,1,\ldots,n-1$$

Αλγόριθμος Πρόσθεσης στο Επίπεδο του Δυαδικού Ψηφίου

$$S = A + B$$

$$A = \sum_{i=0}^{n-1} a_i 2^i, B = \sum_{i=0}^{n-1} b_i 2^i \quad a_i, b_i \in \{0,1\}$$

Λειτουργία την οποία υλοποιεί ένας πλήρης αθροιστής 1-bit

$$S = \sum_{i=0}^{n-1} s_i 2^i$$

$$s_i = a_i \oplus b_i \oplus c_i$$

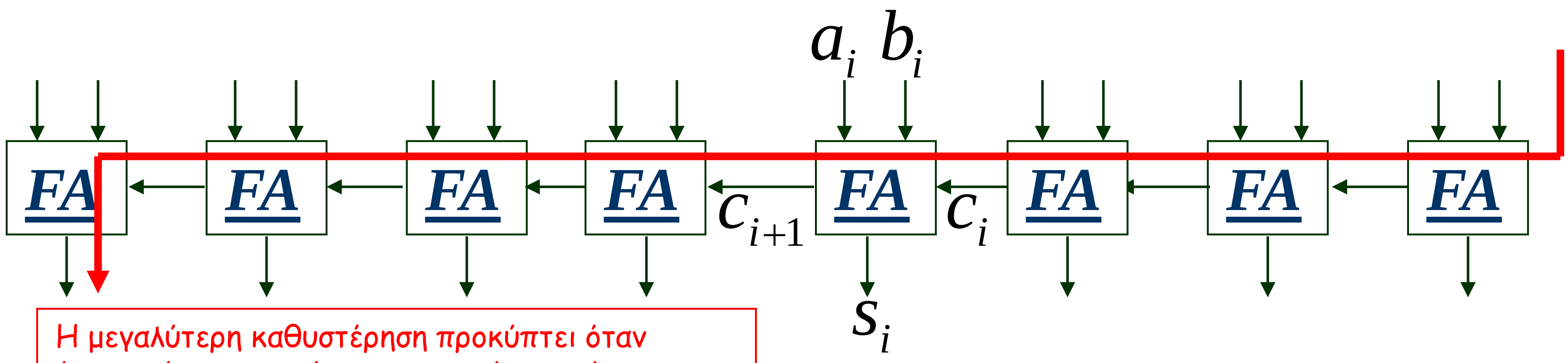
$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

$$i = 0, 1, \ldots, n-1$$

Περίγραμμα Παρουσίασης

- Εισαγωγή
- Αθροιστές διάδοσης κρατουμένου
- Αθροιστές παράλληλου προθέματος
- Άθροιση πολλών εισόδων

Αθροιστής Κυματισμού Κρατουμένου (Ripple Carry)



Η μεγαλύτερη καθυστέρηση προκύπτει όταν ένα ψηφίο κρατουμένου προκαλεί αλλαγές σε όλα τα ψηφία μέχρις και το περισσότερο σημαντικό του αποτελέσματος.

Καθυστέρηση ενός FA

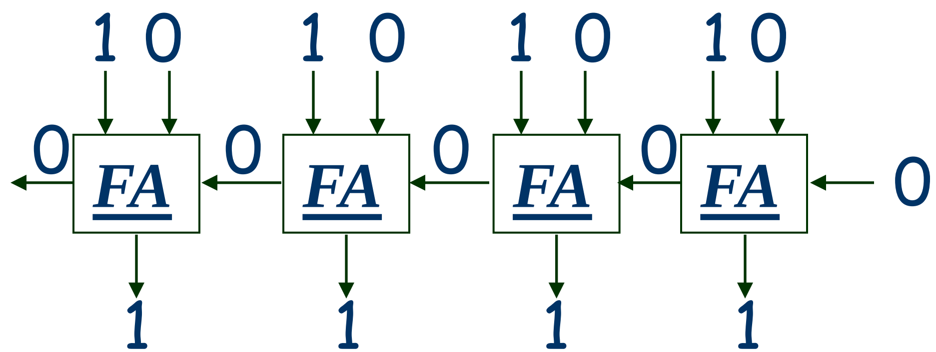
$$A = nA_{\mathrm{FA}} \qquad T_d \approx nT_{\mathrm{fa}}$$

Εμβαδό ενός FA

Η πολυπλοκότητα της υλοποίησης εξαρτάται γραμμικά από το μήκος λέξης *n* των όρων.

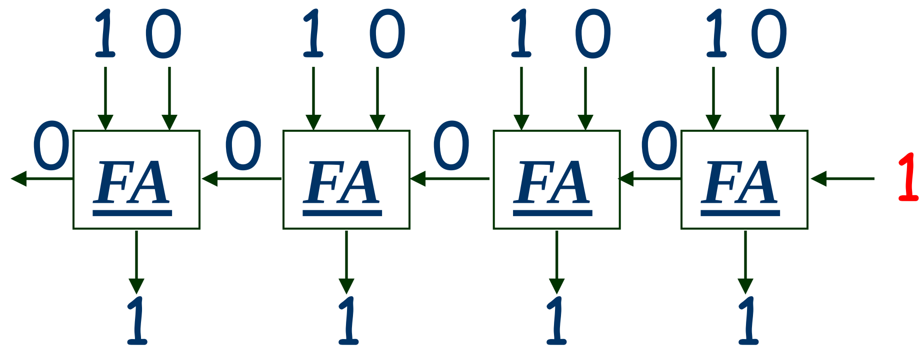
Πλήθος FAs

Μεγαλύτερη Καθυστέρηση σε Αθροιστή 4 Δυαδικών Ψηφίων



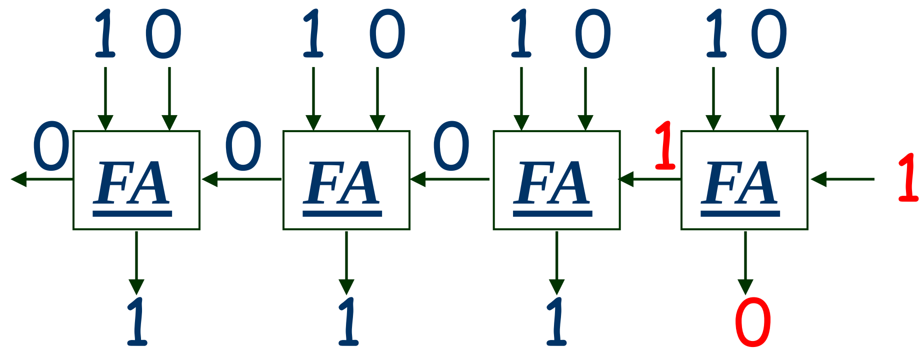
- Θεωρείστε ότι θέλουμε να προσθέσουμε τους δυαδικούς 1111 και 0000 με κρατούμενο εισόδου 0.
- Μετά από ικανό χρονικό διάστημα τα ενδιάμεσα κρατούμενα γίνονται 0 και η έξοδος 1111.
- Έστω τώρα ότι το κρατούμενο εισόδου γίνεται 1.
- Μετά από χρόνο ίσο με την χρονική καθυστέρηση ενός FA (T_{FA}), το κρατούμενο εξόδου του λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1110.
- Μετά από $2T_{FA}$, το κρατούμενο εξόδου του δεύτερου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1100.
- Μετά από $3T_{FA}$, το κρατούμενο εξόδου του τρίτου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1000.
- Μετά από $4T_{FA}$, το κρατούμενο εξόδου του περισσότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 0000.

Μεγαλύτερη Καθυστέρηση σε Αθροιστή 4 Δυαδικών Ψηφίων



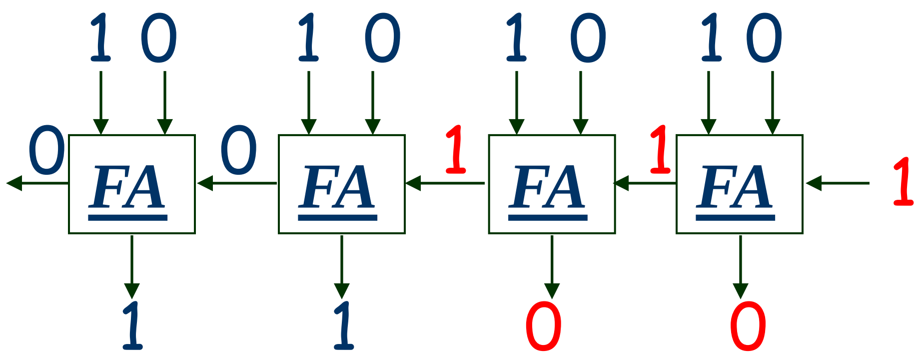
- Θεωρείστε ότι θέλουμε να προσθέσουμε τους δυαδικούς 1111 και 0000 με κρατούμενο εισόδου 0.
- Μετά από ικανό χρονικό διάστημα τα ενδιάμεσα κρατούμενα γίνονται 0 και η έξοδος 1111.
- Έστω τώρα ότι το κρατούμενο εισόδου γίνεται 1.
- Μετά από χρόνο ίσο με την χρονική καθυστέρηση ενός FA (T_{FA}), το κρατούμενο εξόδου του λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1110.
- Μετά από $2T_{FA}$, το κρατούμενο εξόδου του δεύτερου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1100.
- Μετά από $3T_{FA}$, το κρατούμενο εξόδου του τρίτου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1000.
- Μετά από $4T_{FA}$, το κρατούμενο εξόδου του περισσότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 0000.

Μεγαλύτερη Καθυστέρηση σε Αθροιστή 4 Δυαδικών Ψηφίων



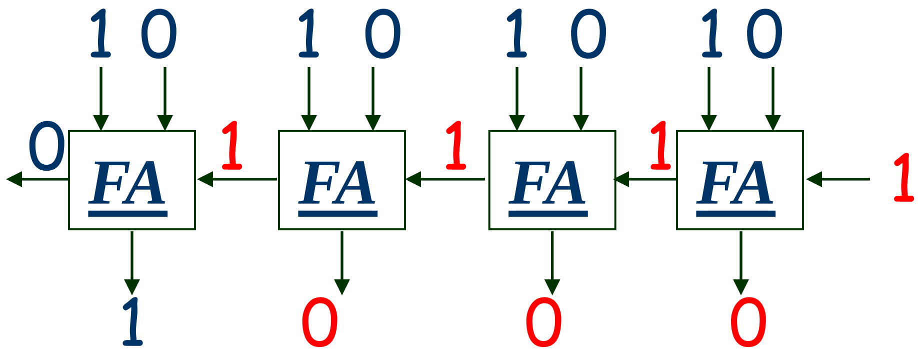
- Θεωρείστε ότι θέλουμε να προσθέσουμε τους δυαδικούς 1111 και 0000 με κρατούμενο εισόδου 0.
- Μετά από ικανό χρονικό διάστημα τα ενδιάμεσα κρατούμενα γίνονται 0 και η έξοδος 1111.
- Έστω τώρα ότι το κρατούμενο εισόδου γίνεται 1.
- Μετά από χρόνο ίσο με την χρονική καθυστέρηση ενός FA (T_{FA}), το κρατούμενο εξόδου του λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1110.
- Μετά από $2T_{FA}$, το κρατούμενο εξόδου του δεύτερου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1100.
- Μετά από $3T_{FA}$, το κρατούμενο εξόδου του τρίτου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1000.
- Μετά από $4T_{FA}$, το κρατούμενο εξόδου του περισσότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 0000.

Μεγαλύτερη Καθυστέρηση σε Αθροιστή 4 Δυαδικών Ψηφίων



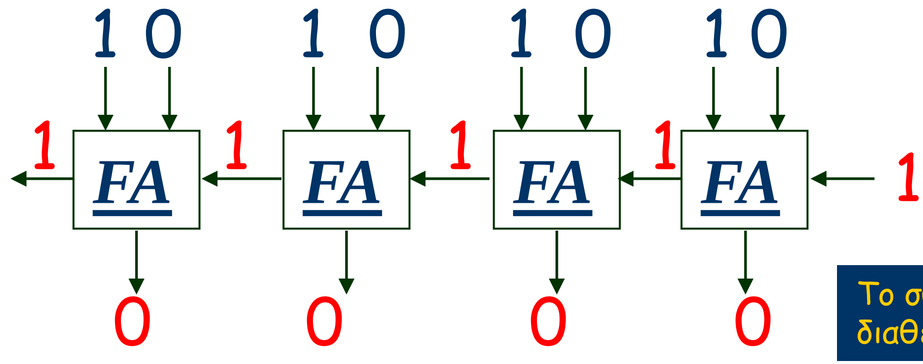
- Θεωρείστε ότι θέλουμε να προσθέσουμε τους δυαδικούς 1111 και 0000 με κρατούμενο εισόδου 0.
- Μετά από ικανό χρονικό διάστημα τα ενδιάμεσα κρατούμενα γίνονται 0 και η έξοδος 1111.
- Έστω τώρα ότι το κρατούμενο εισόδου γίνεται 1.
- Μετά από χρόνο ίσο με την χρονική καθυστέρηση ενός FA (T_{FA}), το κρατούμενο εξόδου του λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1110.
- Μετά από $2T_{FA}$, το κρατούμενο εξόδου του δεύτερου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1100.
- Μετά από $3T_{FA}$, το κρατούμενο εξόδου του τρίτου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1000.
- Μετά από $4T_{FA}$, το κρατούμενο εξόδου του περισσότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 0000.

Μεγαλύτερη Καθυστέρηση σε Αθροιστή 4 Δυαδικών Ψηφίων



- Θεωρείστε ότι θέλουμε να προσθέσουμε τους δυαδικούς 1111 και 0000 με κρατούμενο εισόδου 0.
- Μετά από ικανό χρονικό διάστημα τα ενδιάμεσα κρατούμενα γίνονται 0 και η έξοδος 1111.
- Έστω τώρα ότι το κρατούμενο εισόδου γίνεται 1.
- Μετά από χρόνο ίσο με την χρονική καθυστέρηση ενός FA (T_{FA}), το κρατούμενο εξόδου του λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1110.
- Μετά από $2T_{FA}$, το κρατούμενο εξόδου του δεύτερου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1100.
- Μετά από $3T_{FA}$, το κρατούμενο εξόδου του τρίτου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1000.
- Μετά από $4T_{FA}$, το κρατούμενο εξόδου του περισσότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 0000.

Μεγαλύτερη Καθυστέρηση σε Αθροιστή 4 Δυαδικών Ψηφίων



Το σωστό αποτέλεσμα είναι διαθέσιμο μετά από (T_{FA})

- Θεωρείστε ότι θέλουμε να προσθέσουμε τους δυαδικούς 1111 και 0000 με κρατούμενο εισόδου 0.
- Μετά από ικανό χρονικό διάστημα τα ενδιάμεσα κρατούμενα γίνονται 0 και η έξοδος 1111.
- Έστω τώρα ότι το κρατούμενο εισόδου γίνεται 1.
- Μετά από χρόνο ίσο με την χρονική καθυστέρηση ενός FA (T_{FA}), το κρατούμενο εξόδου του λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1110.
- Μετά από $2T_{FA}$, το κρατούμενο εξόδου του δεύτερου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1100.
- Μετά από $3T_{FA}$, το κρατούμενο εξόδου του τρίτου λιγότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 1000.
- Μετά από $4T_{FA}$, το κρατούμενο εξόδου του περισσότερο σημαντικού αθροιστή γίνεται 1, και το ψηφίο αθροίσματος 0. Το συνολικό αποτέλεσμα είναι τώρα 0000.

Επιτάχυνση Διάδοσης του Κρατουμένου

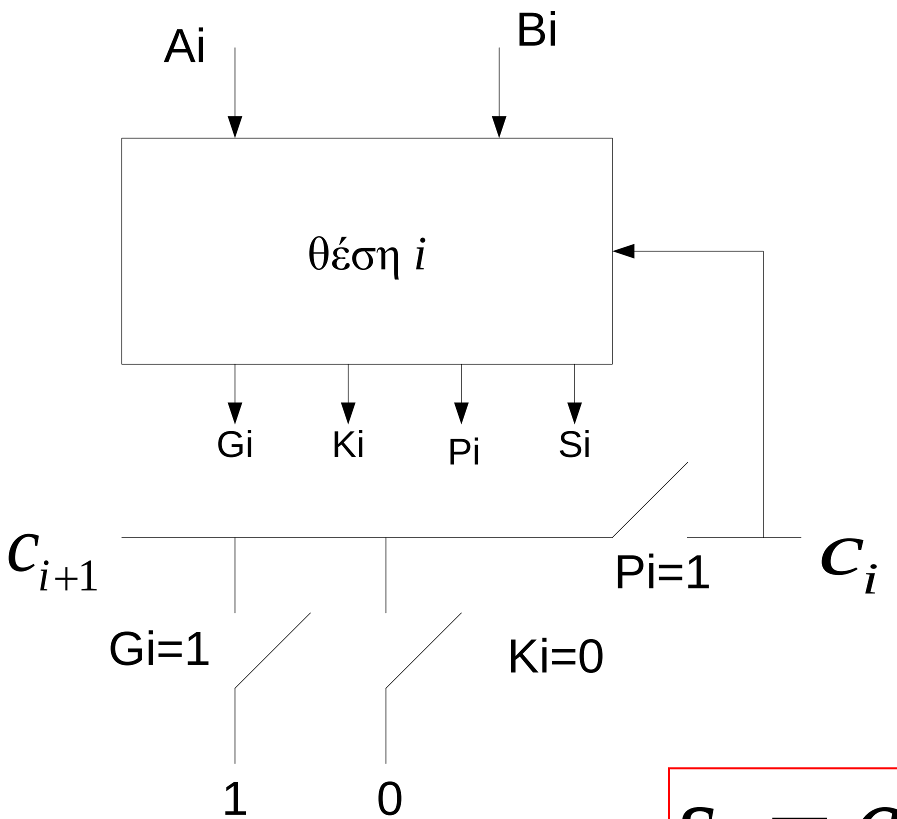
$$c_{i+1} = a_i b_i + a_i c_i + b_i c_i$$
$$= a_i b_i + \left(a_i + b_i\right) c_i$$

- Το ψηφίο κρατουμένου εκφράζεται συναρτήσει νέων σημάτων ελέγχου.

$$\left.\begin{aligned} p_i &= a_i + b_i \\ g_i &= a_i b_i \end{aligned}\right\} \Rightarrow c_{i+1} = g_i + p_i c_i$$

$$k_i = \overline{a_i}\,\overline{b_i}$$

Η Τεχνική Διάδοσης Κρατουμένου Manchester Carry Chain



$$\boxed{s_i = a_i \oplus b_i \oplus c_i}$$

Η Τεχνική Πρόβλεψης Κρατουμένου (carry look ahead)

Ξεκινώντας από τον αναδρομικό ορισμό του κρατούμενου, εκτελούμε *unrolling*

$$c_{i+1} = g_i + p_i c_i$$

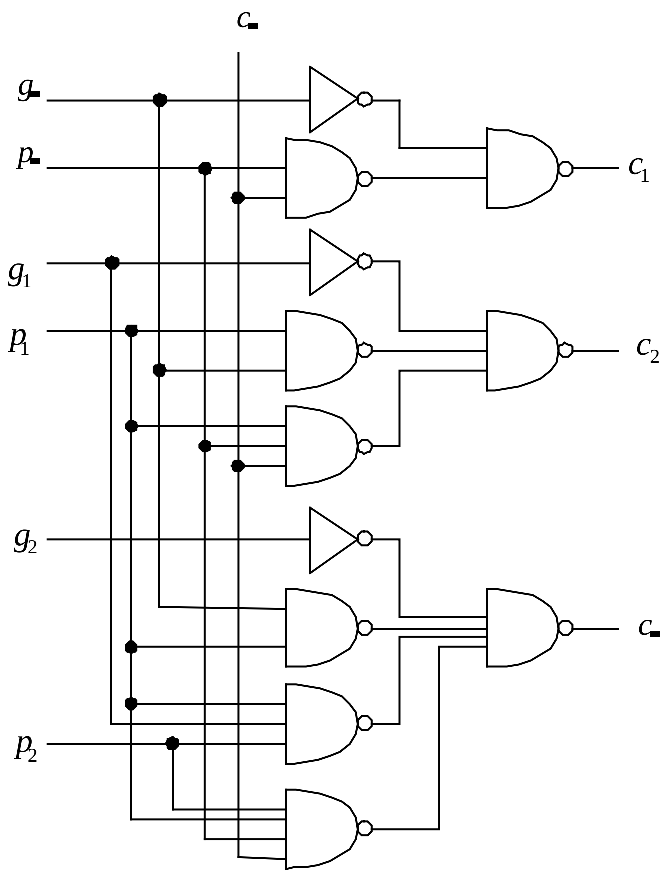
$$c_1 = g_0 + p_0 c_0$$

$$c_2 = g_1 + p_1 c_1 = g_1 + p_1 \left(g_0 + p_0 c_0 \right) = g_1 + p_1 g_0 + p_1 p_0 c_0$$

$$\begin{aligned} c_3 &= g_2 + p_2 c_2 = g_2 + p_2 \left(g_1 + p_1 g_0 + p_1 p_0 c_0 \right) \\ &= g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_0 \\ &\vdots \end{aligned}$$

$$c_{i+1} = g_i + \sum_{j=0}^{i-1} \left(\prod_{k=j+1}^{i} p_k \right) g_j + \prod_{k=0}^{i} p_k c_0$$

Αθροιστής Πρόβλεψης Κρατουμένου



- $p_{n/2} \rightarrow n(n+2)/4$ πύλες.
- Αριθμός transistors

$$\left(n^3 + 9n^2 + 20n\right)/3$$

[(i+2)(i+5) tr ανά ψηφιακή θέση]

- Καθυστέρηση (με fanin)

$$T_D = t_L\left(0.25n^2 + 4.4n + 33.1\right)$$

$t_{L,}$ καθυστέρηση αντιστροφέα

Group Carry Look-Ahead

$$G_{i+3:i} = g_{i+3} + g_{i+2}\,p_{i+3} + g_{i+1}\,p_{i+2}\,p_{i+3} + g_i\,p_{i+1}\,p_{i+2}\,p_{i+3}$$

$$P_{i+3:i} = p_i\,p_{i+1}\,p_{i+2}\,p_{i+3}$$

Group Carry Look-Ahead

$$G_{i+3:i} = g_{i+3} + g_{i+2}\,p_{i+3} + g_{i+1}\,p_{i+2}\,p_{i+3} + g_i\,p_{i+1}\,p_{i+2}\,p_{i+3}$$

$$P_{i+3:i} = p_i\,p_{i+1}\,p_{i+2}\,p_{i+3}$$

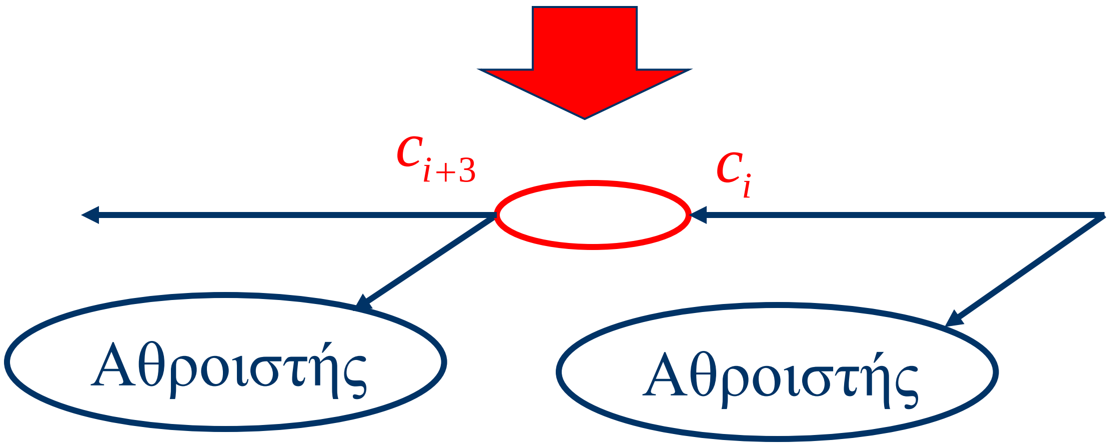
$$c_{i+3} = G_{i+3:i} + P_{i+3:i}c_i$$

Group Carry Look-Ahead

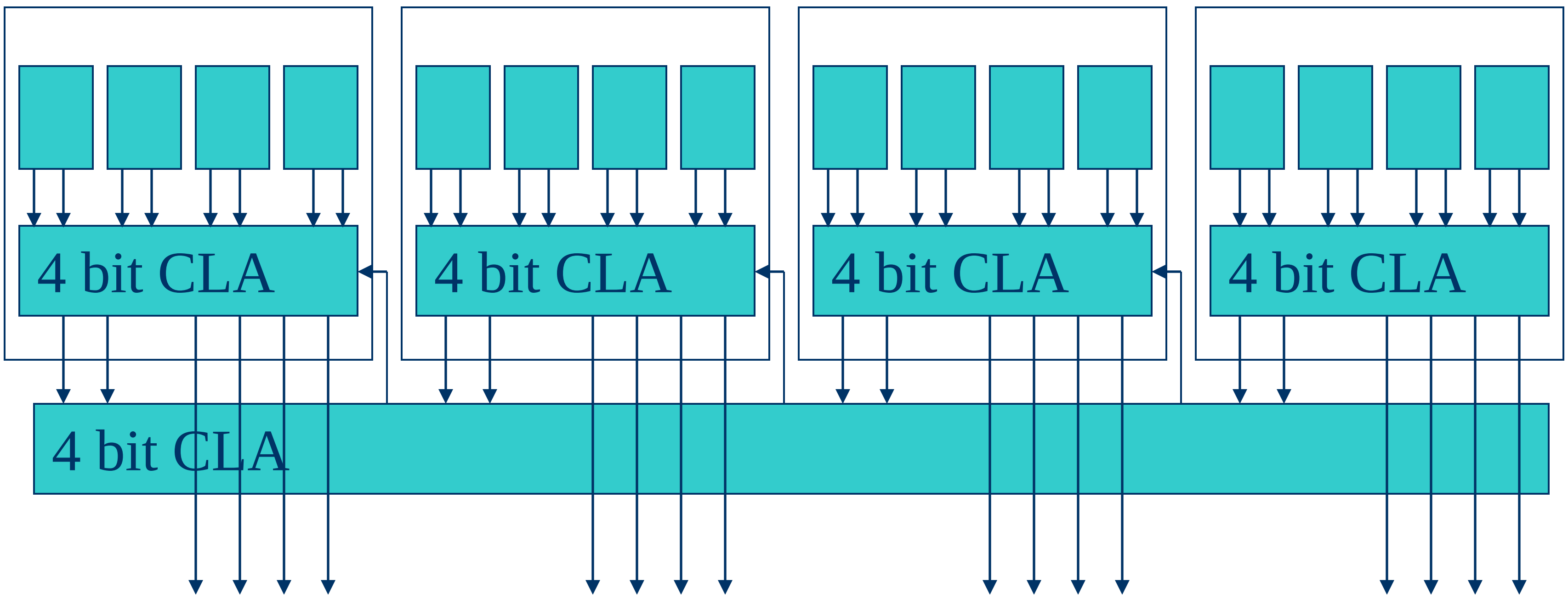
$$G_{i+3:i} = g_{i+3} + g_{i+2}\,p_{i+3} + g_{i+1}\,p_{i+2}\,p_{i+3} + g_i\,p_{i+1}\,p_{i+2}\,p_{i+3}$$

$$P_{i+3:i} = p_i\,p_{i+1}\,p_{i+2}\,p_{i+3}$$

$$c_{i+3} = G_{i+3:i} + P_{i+3:i}c_i$$

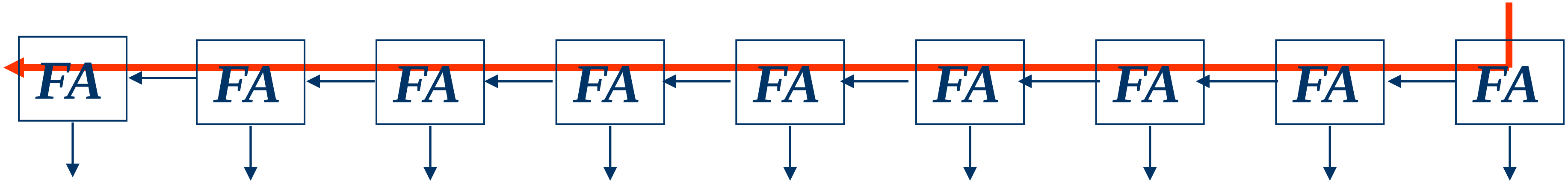


Πρόβλεψη Κρατουμένου Πολλών Επιπέδων

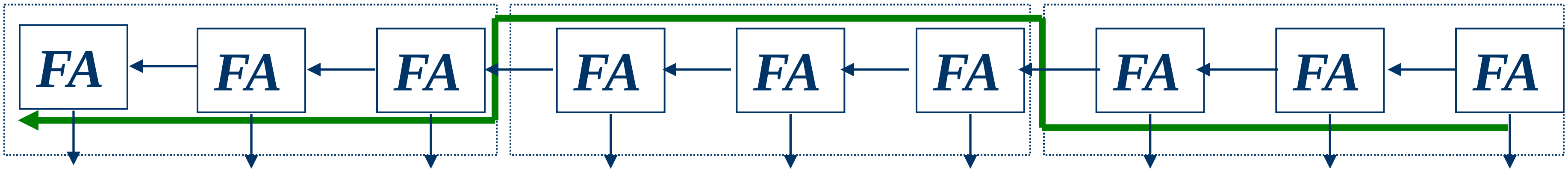


$$T = 4\log_4 n + 1$$

Ο Αθροιστής Carry-skip



Ο Αθροιστής Carry-skip



$$T_d = 2T_{add} + \left(\left\lceil \frac{n}{r} \right\rceil - 2 \right) T_{skip} = 4(r-1) + 2\left(\left\lceil \frac{n}{r} \right\rceil - 2 \right)$$

$$= 4(r-2) + 2\left\lceil \frac{n}{r} \right\rceil$$

n, μήκος λέξης αθροιστή

r, μήκος μπλοκ

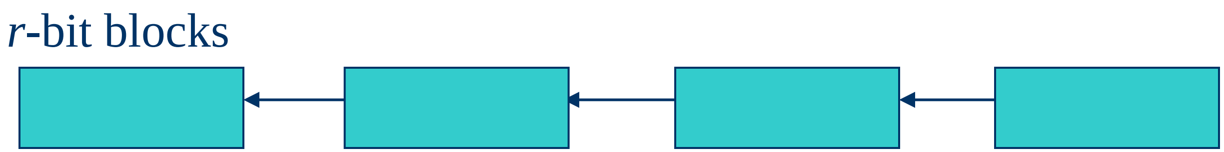
Βέλτιστος Carry-Skip Σταθερού Μήκους Μπλοκ

$$T(r) = 4(r-2) + 2\frac{n}{r}$$

$$\frac{dT}{dr} = 4 - 2\frac{n}{r^2} \qquad \frac{dT}{dr} = 0 \Leftrightarrow r_{\text{opt}} = \sqrt{\frac{n}{2}}$$

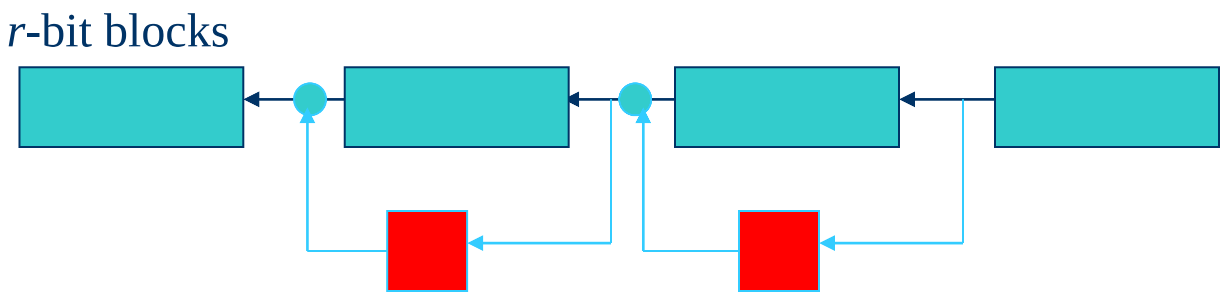
$$\boxed{T_{\text{opt}} = 4\left(\sqrt{\frac{n}{2}} - 2\right) + 2\frac{n}{\sqrt{\frac{n}{2}}} = 4\sqrt{2n} - 8}$$

Αθροιστής Carry-Skip Πολλών Επιπέδων



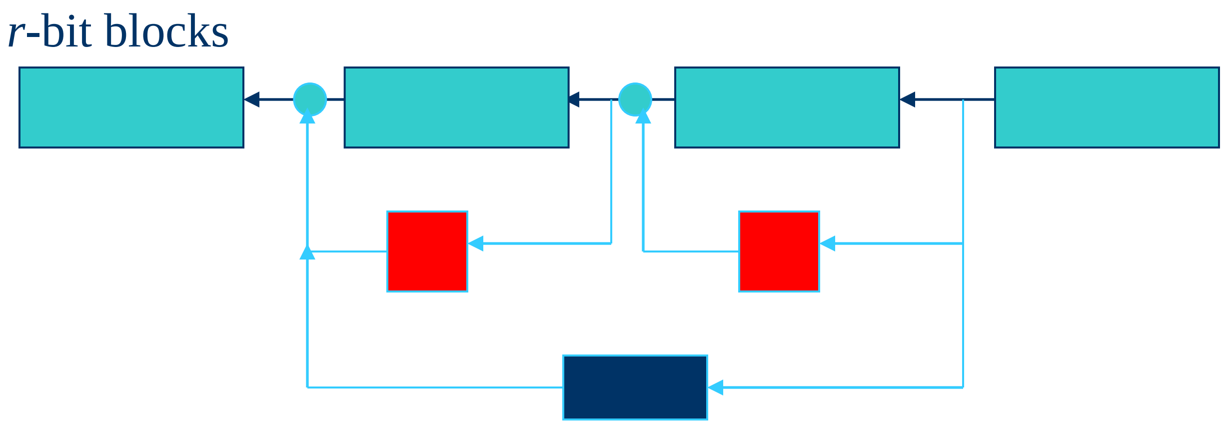
Καταργεί συνεχόμενες παρακάμψεις.

Αθροιστής Carry-Skip Πολλών Επιπέδων



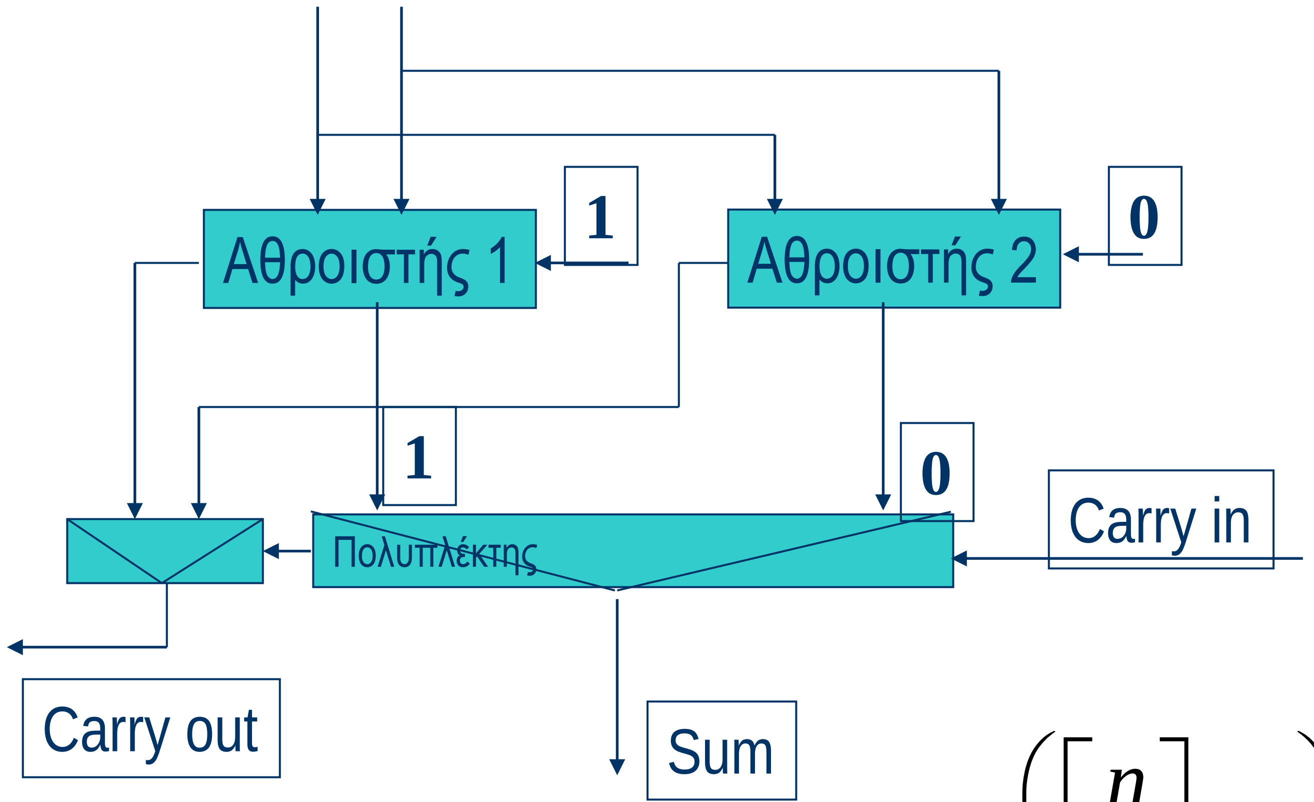
Καταργεί συνεχόμενες παρακάμψεις.

Αθροιστής Carry-Skip Πολλών Επιπέδων



Καταργεί συνεχόμενες παρακάμψεις.

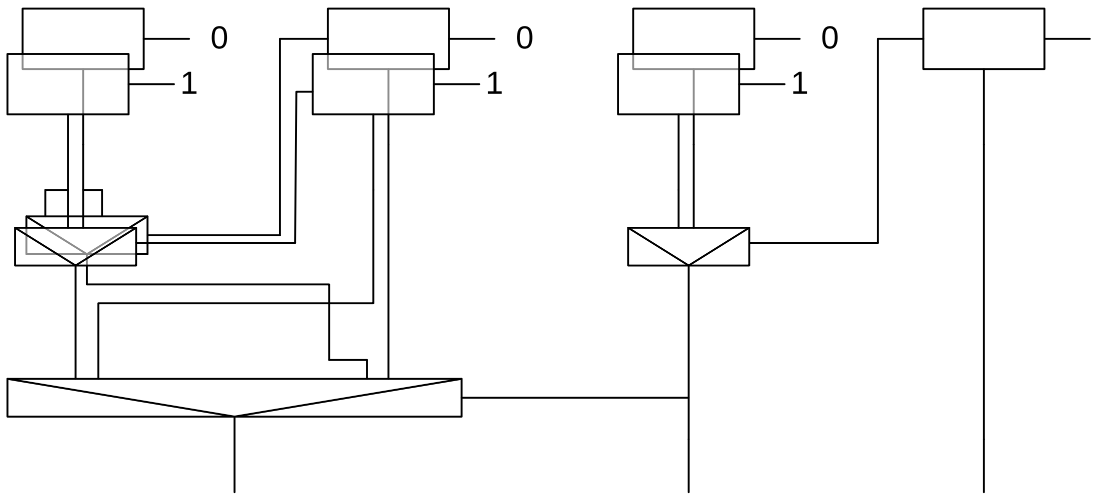
Ο Αθροιστής Επιλογής Κρατουμένου (carry-select)



σειριακή εξάρτηση
carry-out από carry-in

$$T_d = 2\left(\left\lceil \frac{n}{r} \right\rceil - 1\right) + T_{add}(r)$$

Αθροιστής Επιλογής Κρατουμένου Δύο Επιπέδων



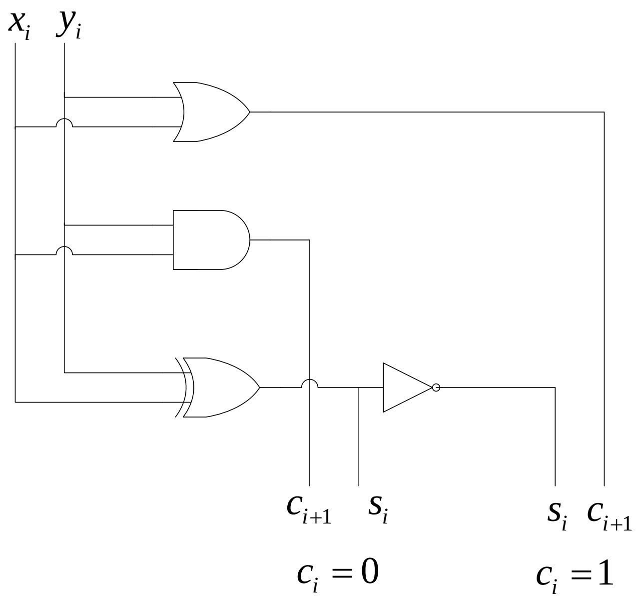
$$T_d = K + 2\left\lceil \log_2\left(\left\lceil \frac{n}{r} \right\rceil - 1\right) \right\rceil$$

Η επιλογή των αποτελεσμάτων γίνεται με δυαδικό δένδρο.

Ο κάθε αθροιστής έχει μήκος $r = n / 4$

Ο Αθροιστής Conditional-Sum

- Ο **carry-select** με αθροιστές μήκους r =1 λέγεται αθροιστής **conditional sum.**
- Το κύκλωμα άθροισης χρησιμοποιεί κοινό hardware για τις δύο περιπτώσεις.



Ο Αθροιστής Ling

- Τροποποίηση των βοηθητικών σημάτων του Carry Look Ahead.

$$h_i = c_i + c_{i-1} \qquad t_i = p_i + g_i$$ (σήμα Transmit)

- Αρχή λειτουργίας αθροιστή Ling

$$\boxed{p_i = a_i \oplus b_i}$$

$$\begin{aligned} c_{i-1}p_{i-1} &= c_{i-1}p_{i-1} + g_{i-1}p_{i-1} + c_{i-1}p_{i-1}p_{i-1} \\ &= c_{i-1}p_{i-1} + \left(g_{i-1} + c_{i-1}p_{i-1}\right)p_{i-1} \\ &= c_{i-1}p_{i-1} + c_i p_{i-1} = \left(c_{i-1} + c_i\right)p_{i-1} = h_i p_{i-1} \end{aligned}$$

$$\begin{aligned} c_i &= g_{i-1} + c_{i-1}p_{i-1} \\ &= h_i g_{i-1} + h_i p_{i-1} = h_i t_{i-1} \end{aligned}$$

$$\begin{aligned} h_i &= c_i + c_{i-1} = \left(g_{i-1} + p_{i-1}c_{i-1}\right) + c_{i-1} \\ &= g_{i-1} + c_{i-1} = g_{i-1} + h_{i-1}t_{i-2} \end{aligned}$$

$$s_i = p_i \oplus c_i = p_i \oplus h_i t_{i-1} = \left(t_i \oplus h_{i+1}\right) + h_i g_i t_{i-1}$$

Το Πλεονέκτημα του Αθροιστή Ling

- Unrolling του διαδιδόμενου σήματος για τέσσερις όρους (Ling):

$$h_i = g_{i-1} + g_{i-2} + g_{i-3}t_{i-2} + g_{i-4}t_{i-3}t_{i-2} + h_{i-4}t_{i-4}t_{i-3}t_{i-2}$$

- Unrolling του διαδιδόμενου σήματος για τέσσερις όρους (CLA):

$$c_i = g_{i-1} + g_{i-2}t_{i-1} + g_{i-3}t_{i-2}t_{i-1} + g_{i-4}t_{i-3}t_{i-2}t_{i-1} + c_{i-4}t_{i-4}t_{i-3}t_{i-2}t_{i-1}$$

- Ένας παράγοντας λιγότερος στους όρους πολλών παραγόντων $\Rightarrow$ μείωση fan-in

Συνδυασμένοι (Hybrid) Αθροιστές

- Οι αθροιστές των επεξεργαστών συνδυάζουν τεχνικές.

- Συνήθως κάποια διασταύρωση carry look ahead και carry select είναι αποδοτικότερη.
  - Τεχνολογία
  - Μήκος λέξης

Ασυμπτωτικές Πολυπλοκότητες Αθροιστών

Είδος Αθροιστή	Σύντμηση	Χρόνος	Εμβαδό
Κυματισμού (ripple)	RCA	$O(n)$	$O(n)$
Manchester	MCC	$O(n)$	$O(n)$
Constant width Carry Skip	CSK	$O(\sqrt{n})$	$O(n)$
Variable width Carry Skip	VSK	$O(\sqrt{n})$	$O(n)$
Carry Select	CSL	$O(\sqrt{n})$	$O(n)$
Carry look ahead	CLA	$O(\log n)$	$O(n \log n)$
ELM	ELM	$O(\log n)$	$O(n \log n)$
Brent-Kung	B&K	$O(\log n)$	$O(n \log n)$
Signed Digit (βάση r)	SD-r	$O(b)$	$O(n)$
Carry-save	CSA	$O(1)$	$O(n)$

Περίγραμμα Παρουσίασης

- Εισαγωγή
- Αθροιστές διάδοσης κρατουμένου
- Αθροιστές παράλληλου προθέματος
- Άθροιση πολλών εισόδων

Γιατί Υπολογισμός Προθέματος (Prefix)

- Το πρόβλημα υπολογισμού του κρατουμένου με τη χρήση βοηθητικών σημάτων μετασχηματίζεται σε ***γνωστό πρόβλημα.***
- Αντιμετωπίζει το σειριακό υπολογισμό του κρατουμένου ⇔ CLA
- Αντιμετωπίζει το πρόβλημα του fan-out στον carry look-ahead.

Κυκλώματα Παράλληλου Προθέματος (Parallel Prefix)

- Υπολογίζεται η παράσταση

$$a_1 \circ a_2 \circ \cdots \circ a_n$$

όπου a_i γνωστά

προσεταιριστικός αλλά όχι αντιμεταθετικός τελεστής

Παράδειγμα:

$$\left(a_1 \circ a_2\right) \circ \left(a_3 \circ a_4\right) \circ \left(a_5 \circ a_6\right) = \left(\left(a_1 \circ a_2\right) \circ a_3\right) \circ \left(\left(a_4 \circ a_5\right) \circ a_6\right)$$

Ο Τελεστής Υπολογισμού Κρατουμένου

$$\boxed{(P,G)\circ(\tilde{P},\tilde{G})\circ(P\cdot\tilde{P},G+P\cdot\tilde{G})}$$

Εφαρμόζεται σε ζεύγη σημάτων propagate, generate

$$P_{i:j}=\begin{cases}P_i, & i=j\\ P_i\cdot P_{i-1:j}, & i>j\end{cases}$$

Ορισμός σημάτων group propagate, generate

$$G_{i:j}=\begin{cases}G_i, & i=j\\ G_i+P_i\cdot G_{i-1:j}, & i>j\end{cases}$$

$$\left(P_{i:j},G_{i:j}\right)=\left(P_{i:m},G_{i:m}\right)\circ\left(P_{m-1:j},G_{m-1:j}\right),\ i\geq m\geq j+1$$

Μη επικαλυπτόμενα groups

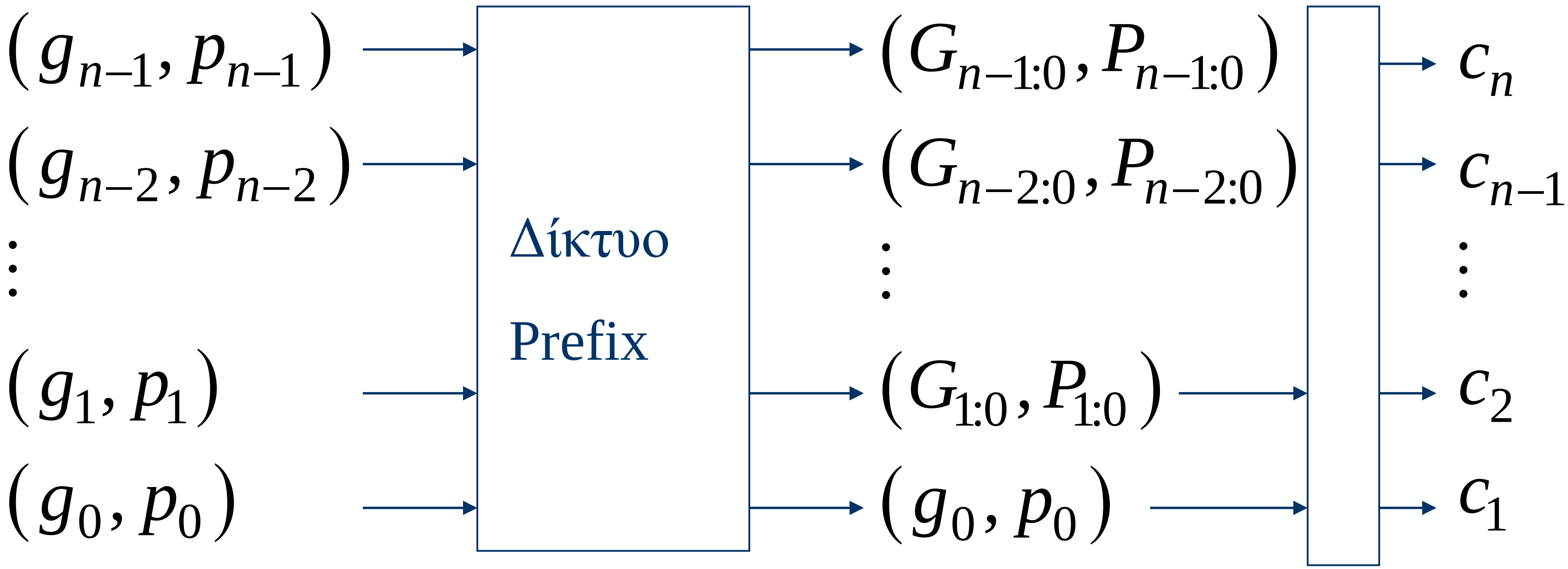
$$\left(P_{i:j},G_{i:j}\right)=\left(P_{i:m},G_{i:m}\right)\circ\left(P_{v:j},G_{v:j}\right),\ i\geq m,\quad v\geq j,\quad v\geq m-1$$

Επικαλυπτόμενα groups

Αρχή Λειτουργίας Αθροιστών Parallel Prefix

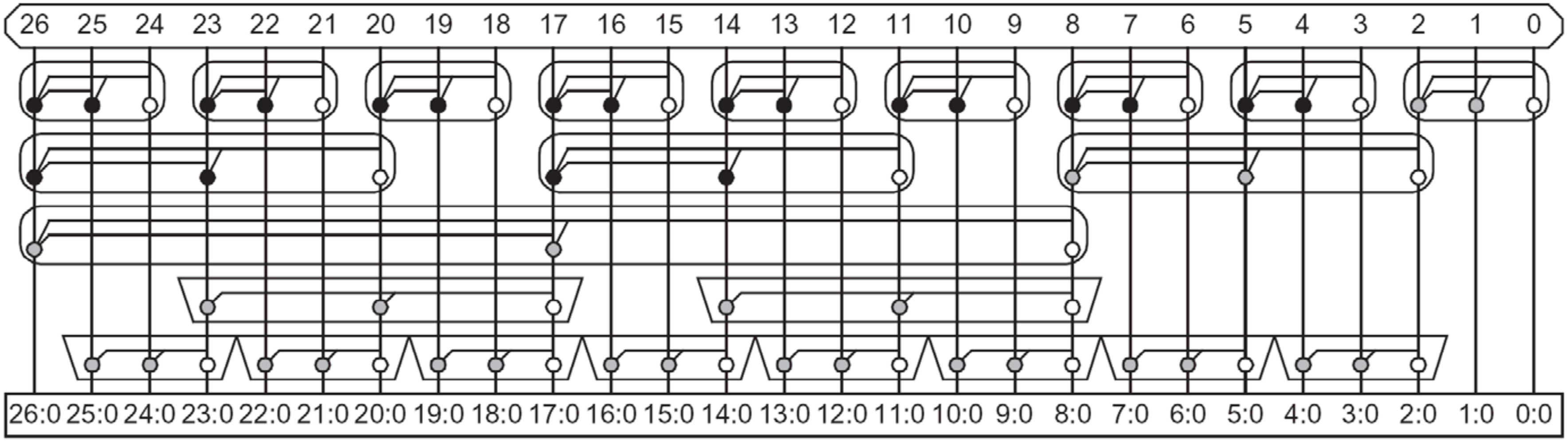
- Πρόβλημα το fanout των σημάτων propagate ανά bit σε CLA
- Λύση ⇒ Χρήση propagate και generate σημάτων από επικαλυπτόμενα groups
- Αποδοτικός υπολογισμός με parallel prefix δίκτυα τελεστή κρατουμένου

Οργάνωση Αθροιστών Parallel Prefix

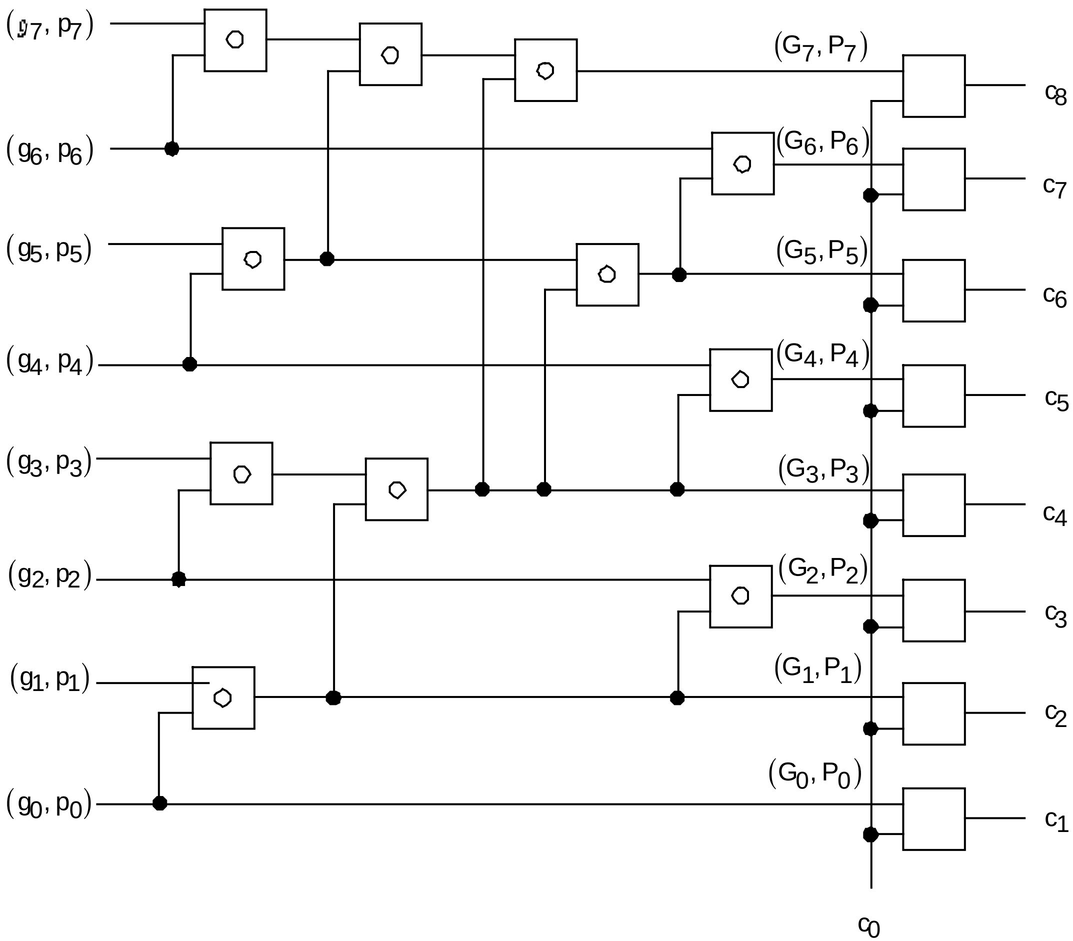


$$\boxed{c_m = G_{m:0} + P_{m:0}c_0, \quad m > 0}$$

Αθροιστής Παράλληλου Προθέματος των Brent-Kung



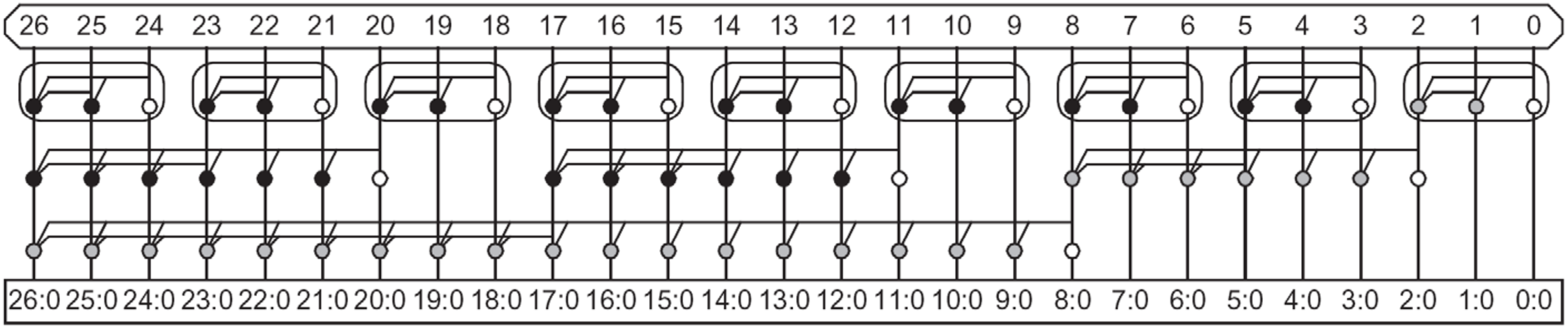
Κρατούμενα σε Brent-Kung



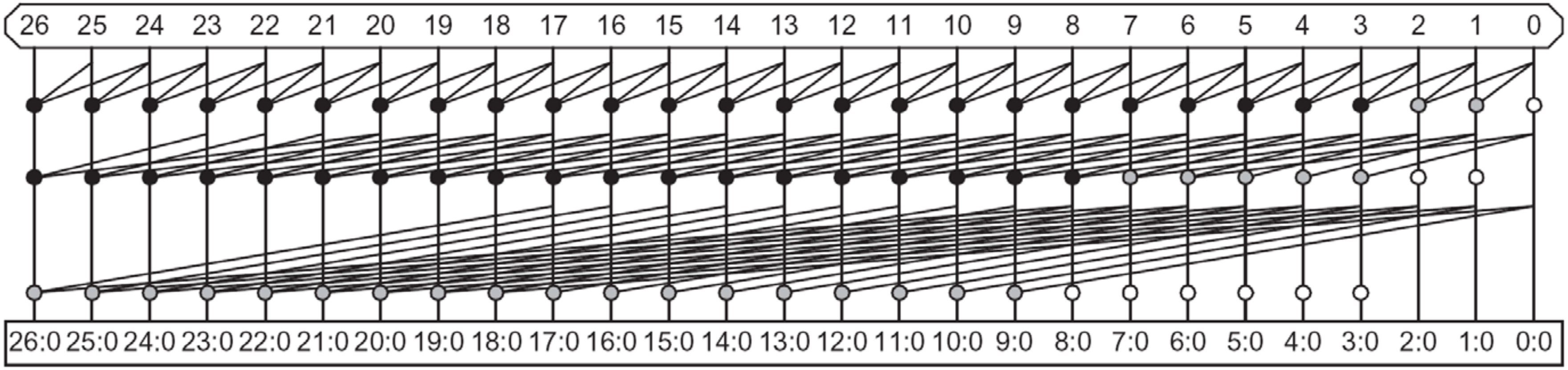
$$N_{tr} = 64n - 16\log_2 n - 32 \quad n = 2^k$$

$$T_{D,ADD} = t_L\left(2\log_2^2 n + 16\log_2 n + 17.8\right)$$

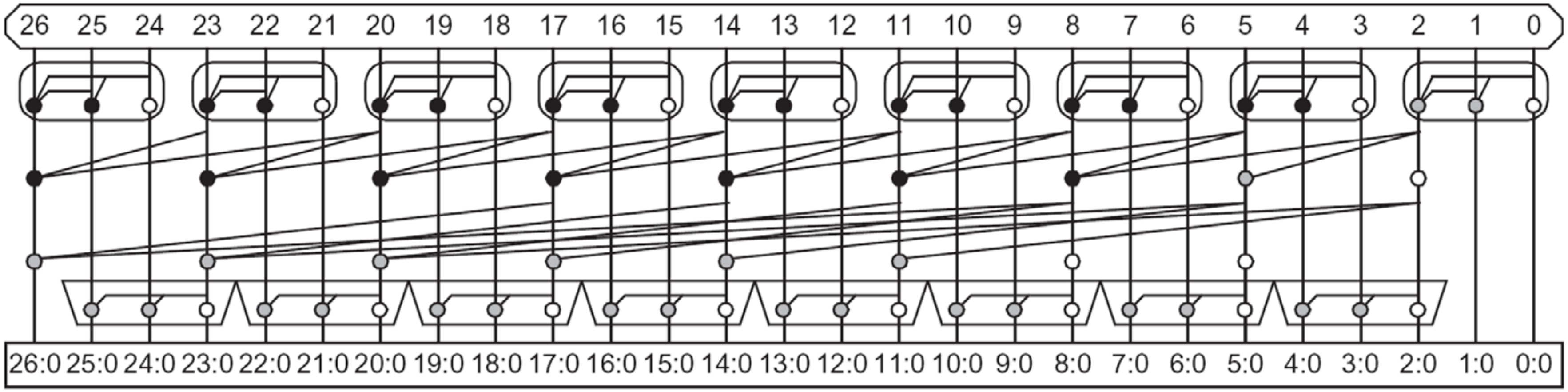
Αθροιστής Ladner-Fischer



Αθροιστής Kogge-Stone



Αθροιστής Han-Carlson



Περίγραμμα Παρουσίασης

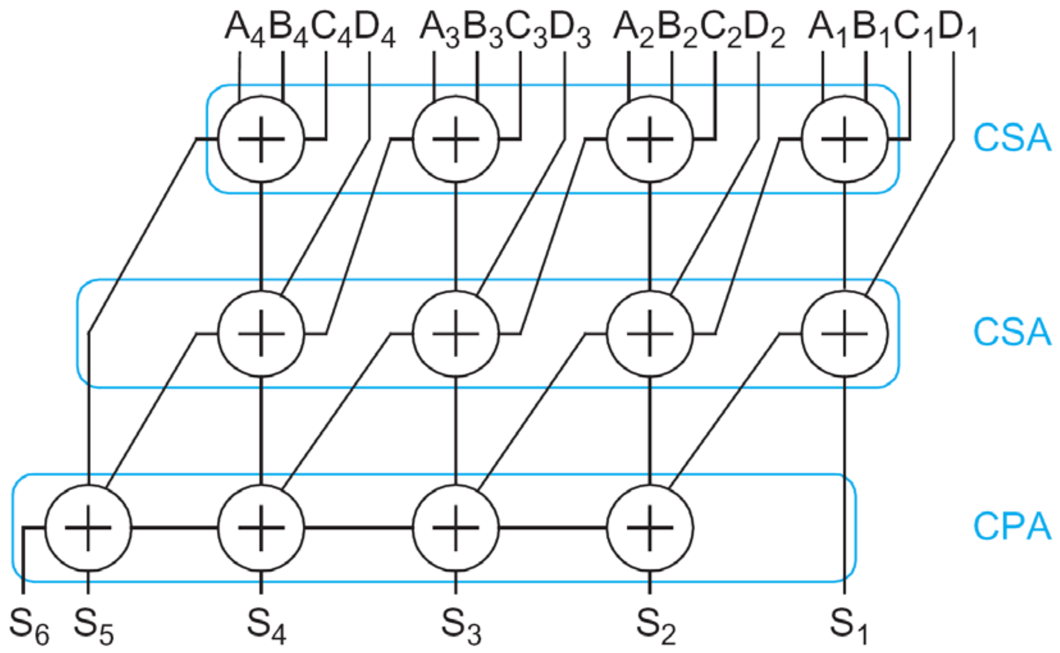
- Εισαγωγή

- Αθροιστές διάδοσης κρατουμένου

- Αθροιστές παράλληλου προθέματος

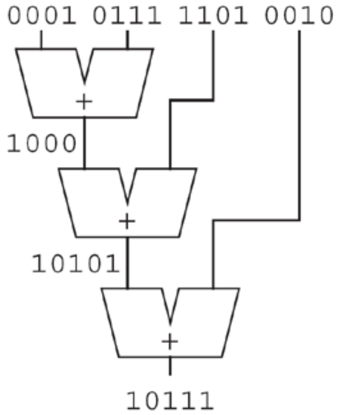
- Άθροιση πολλών εισόδων

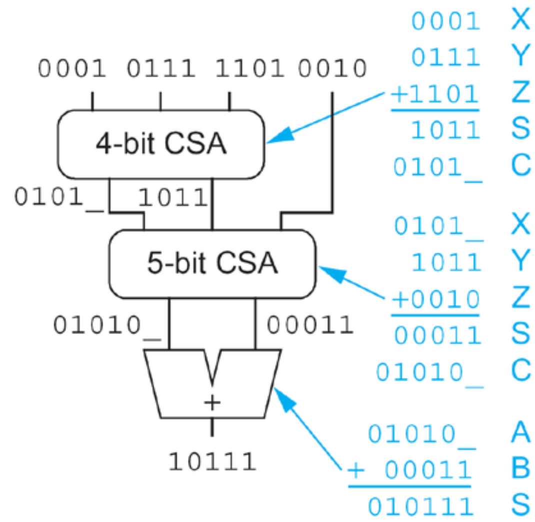
Άθροιση Διατήρησης Κρατουμένου (carry save)



$$T = (k-2)T_{CSA} + T_{CPA}$$

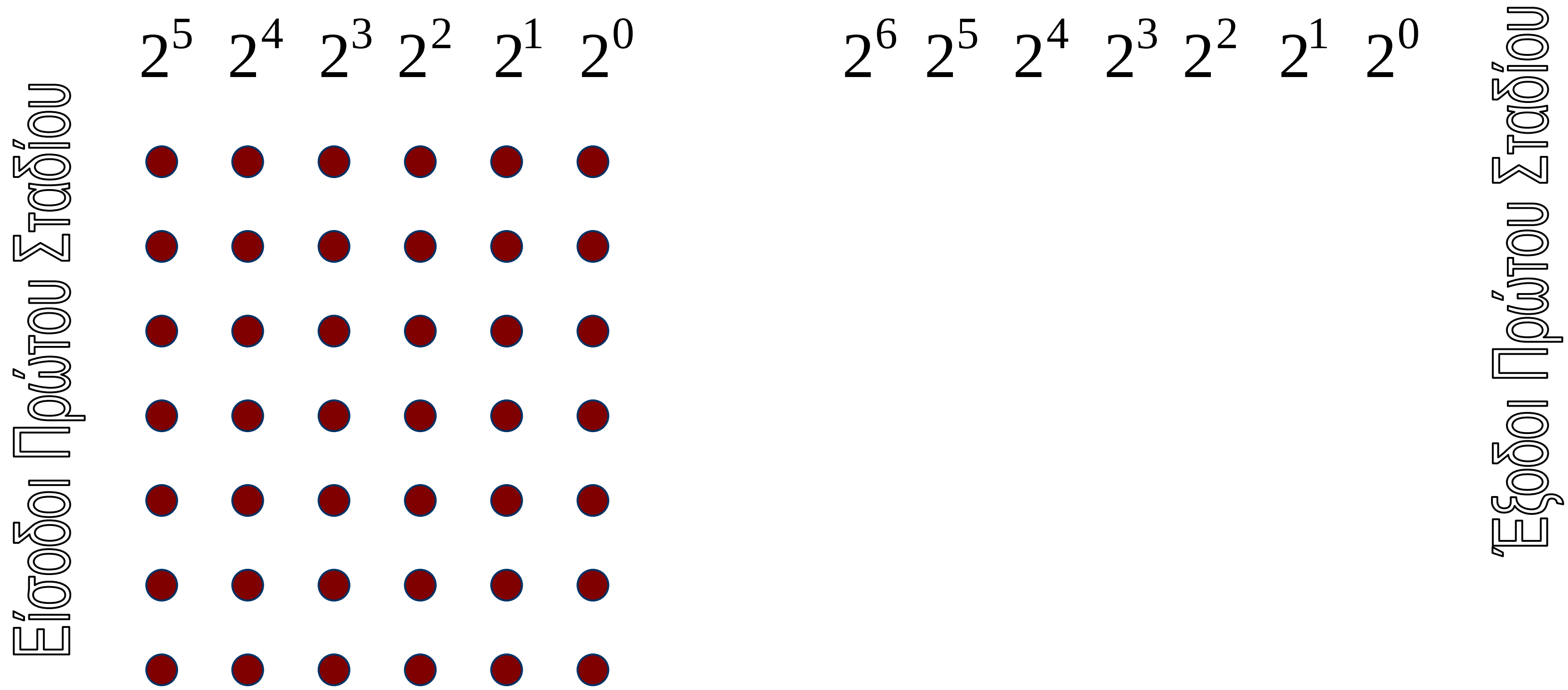
Δένδρα Carry Save Αθροιστών





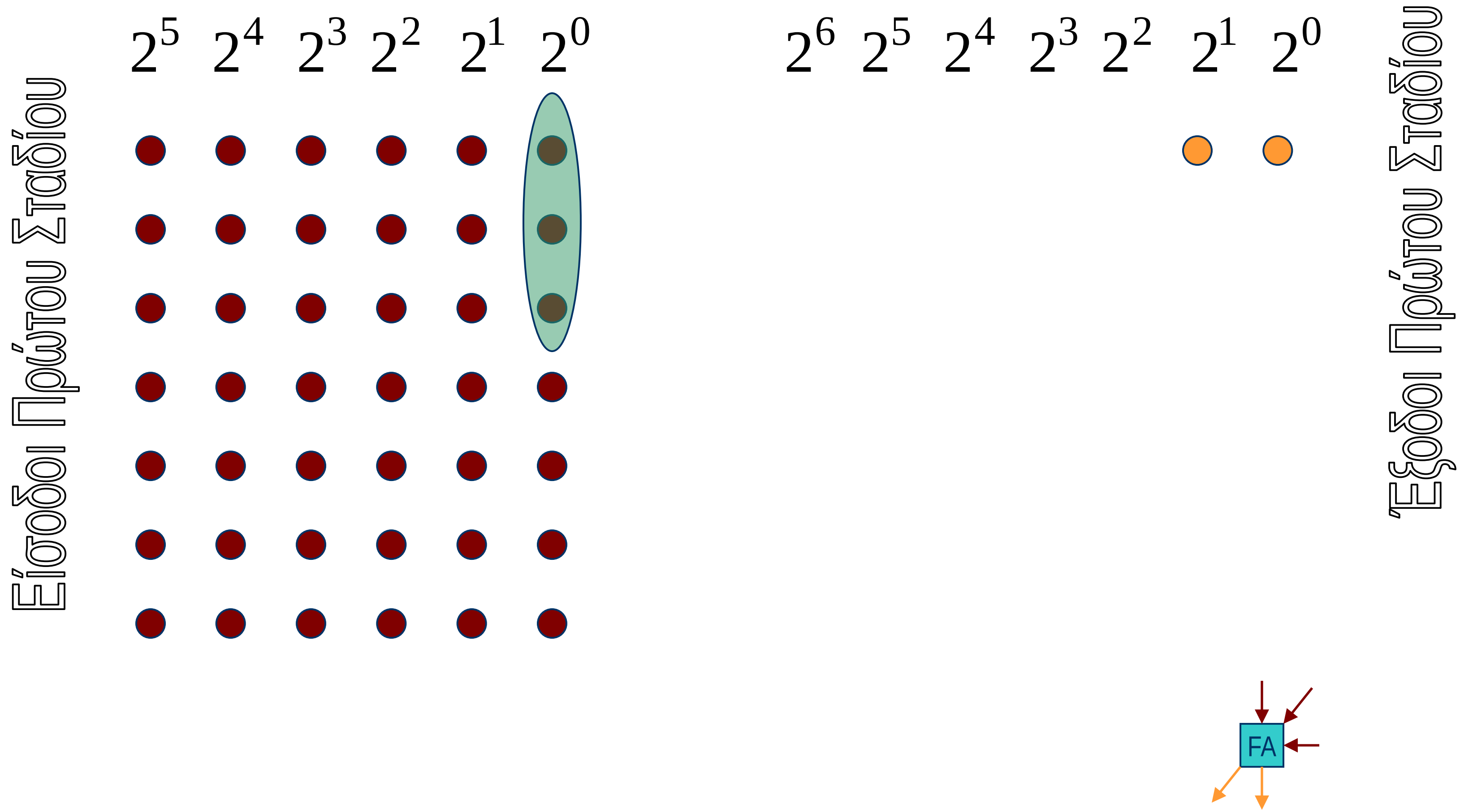
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



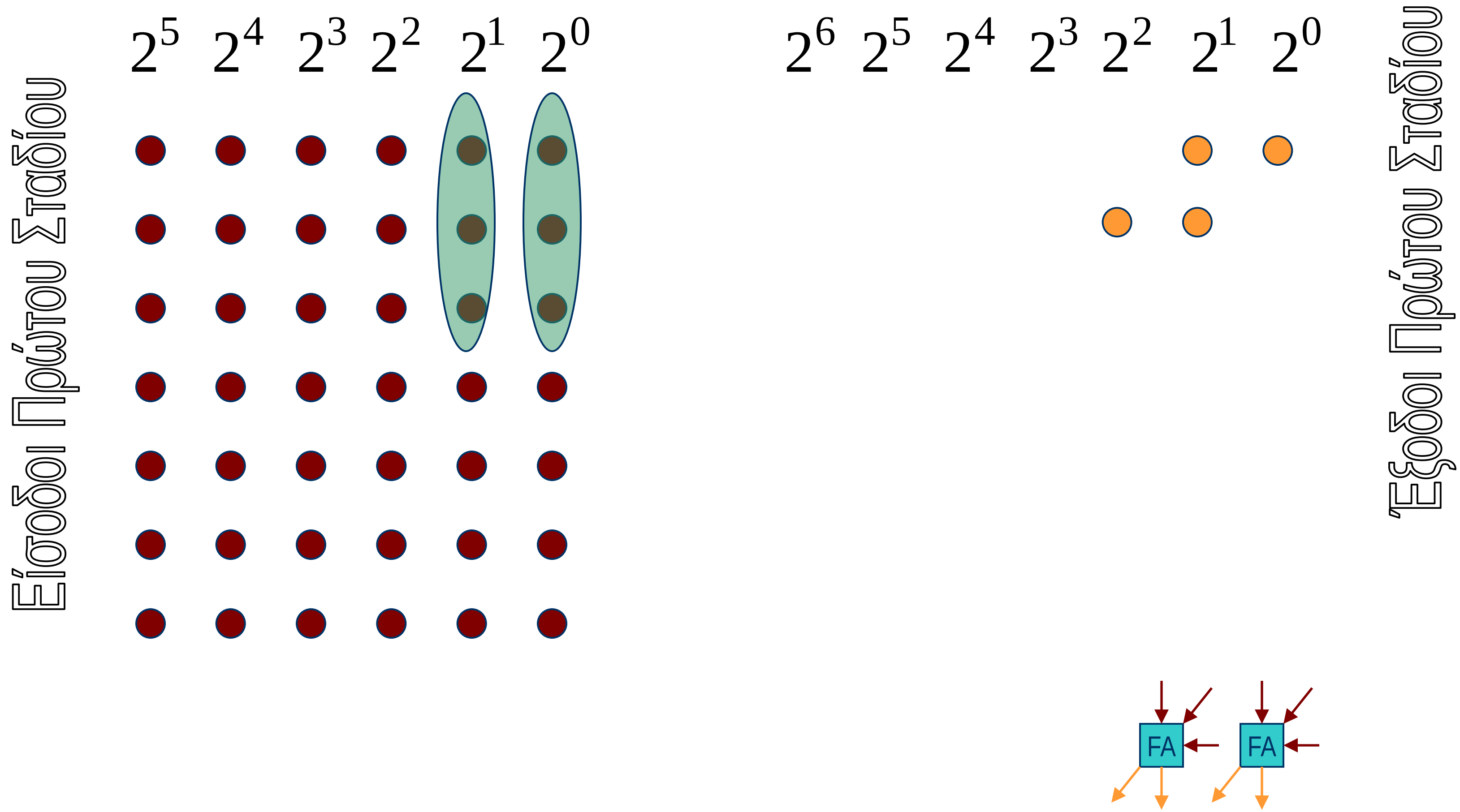
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



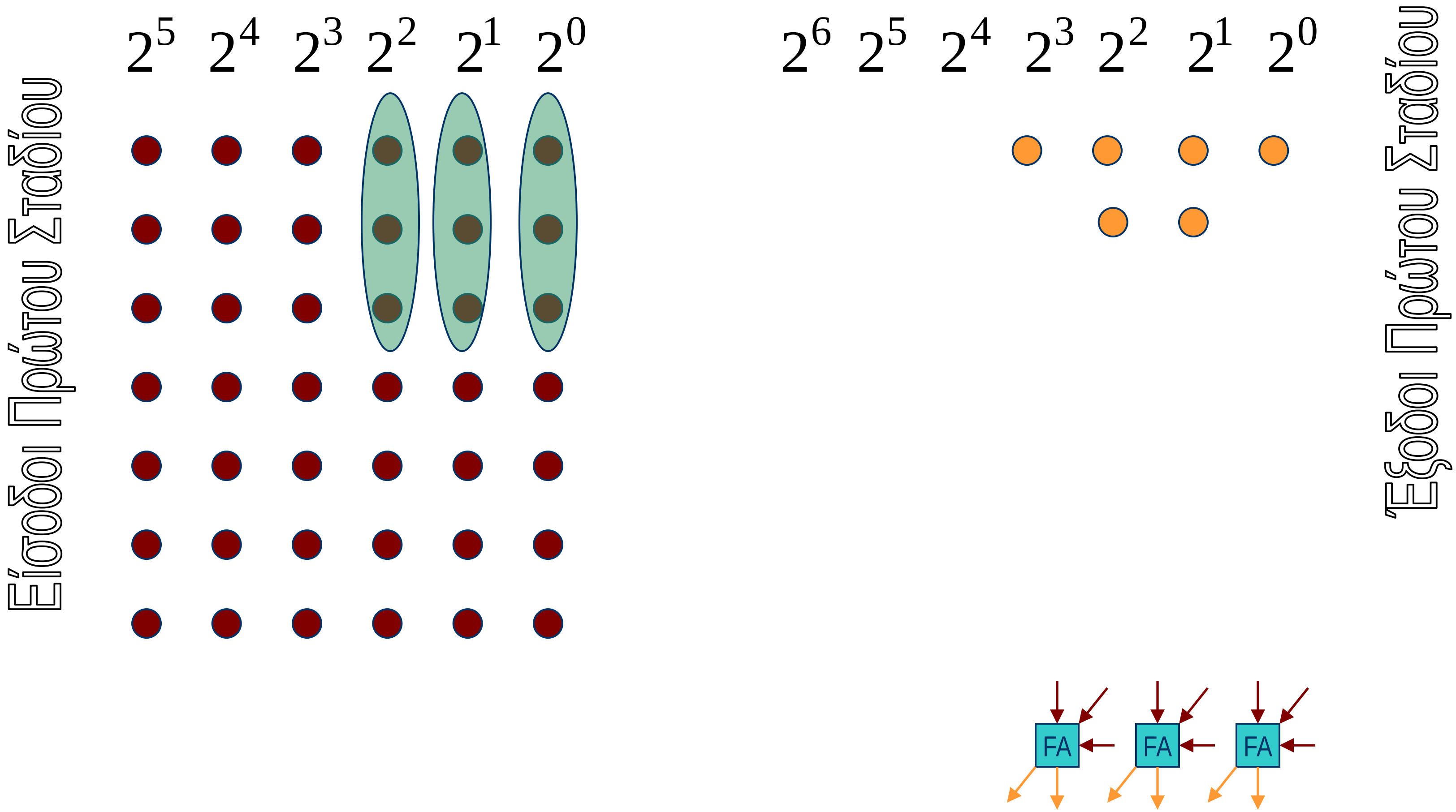
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



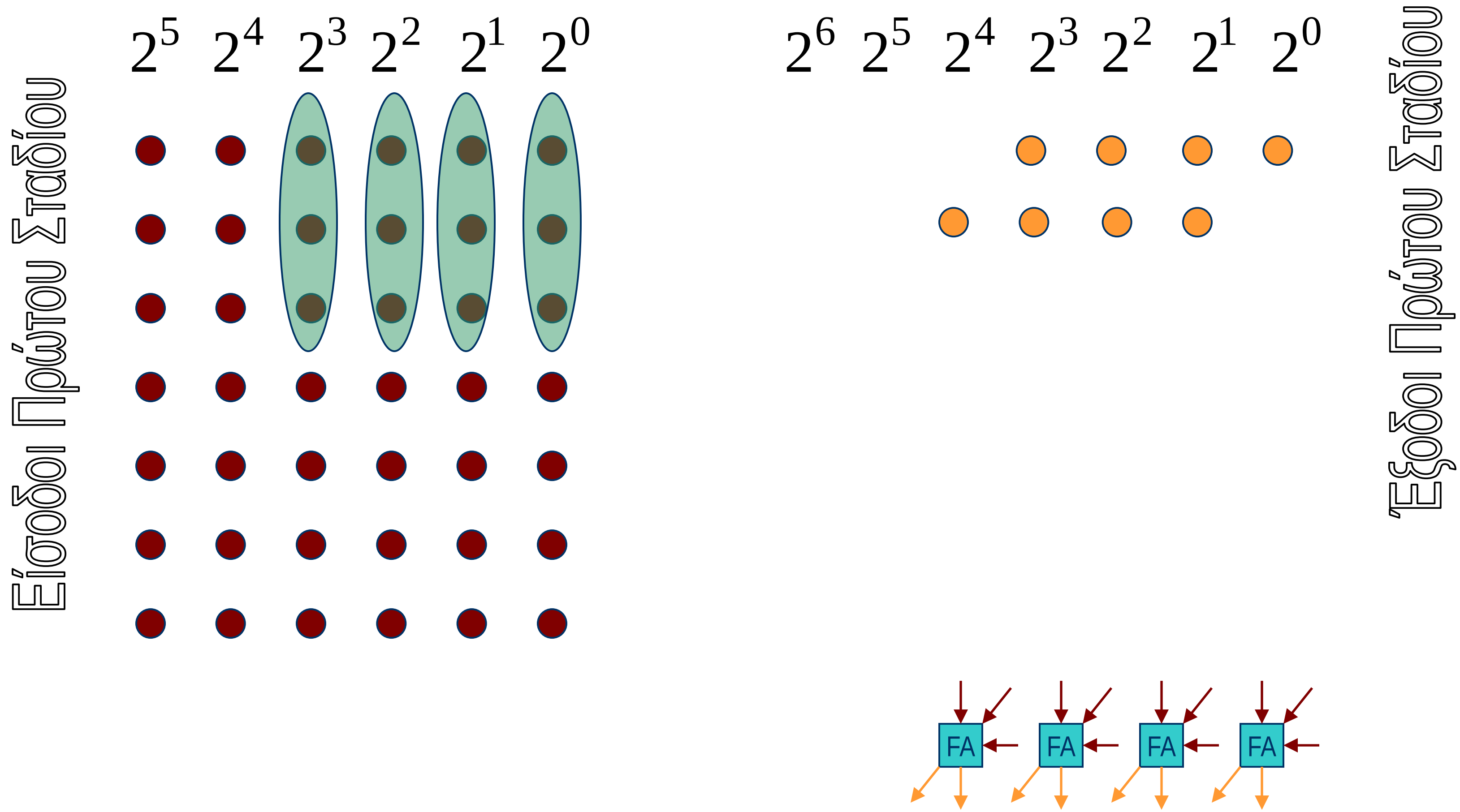
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



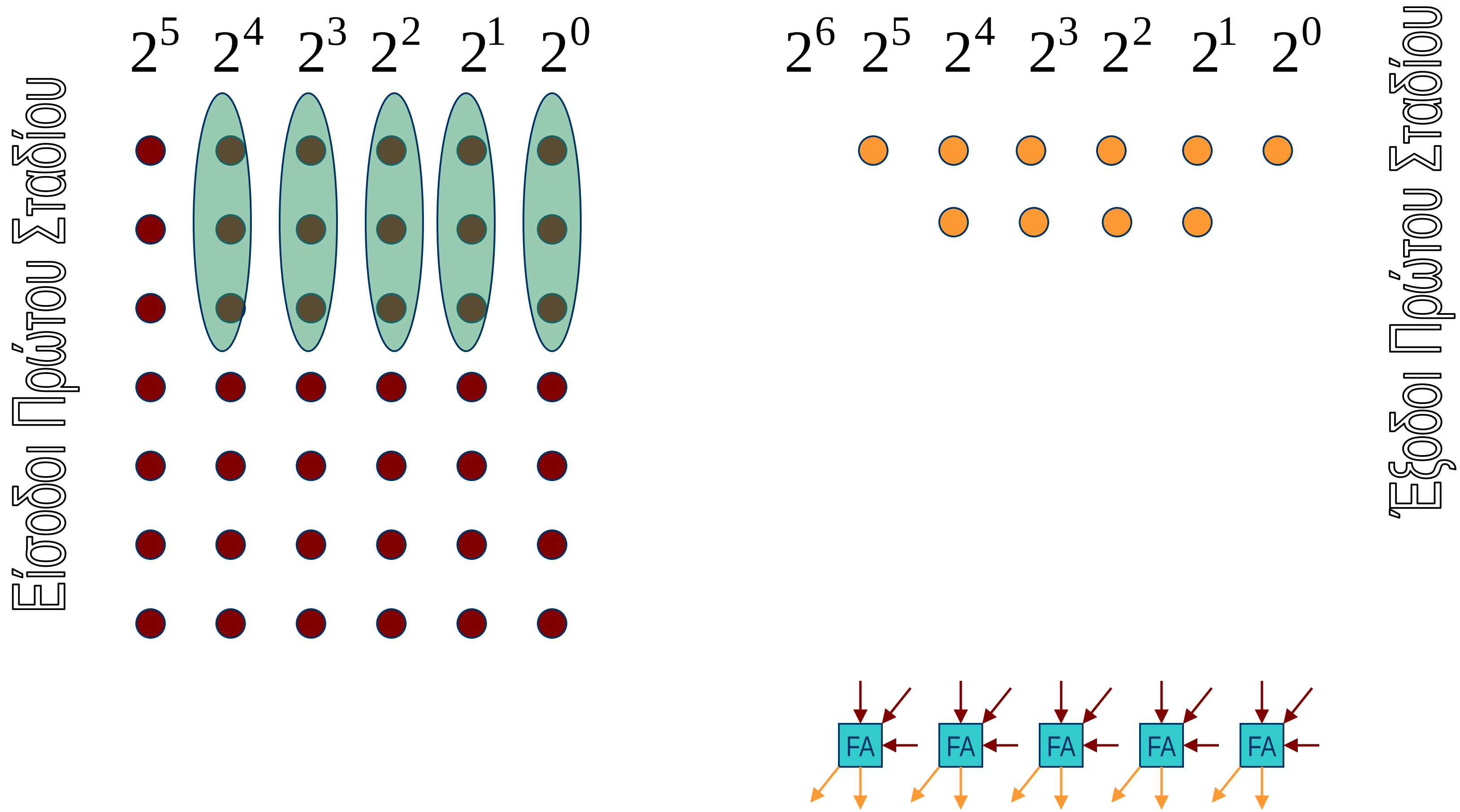
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



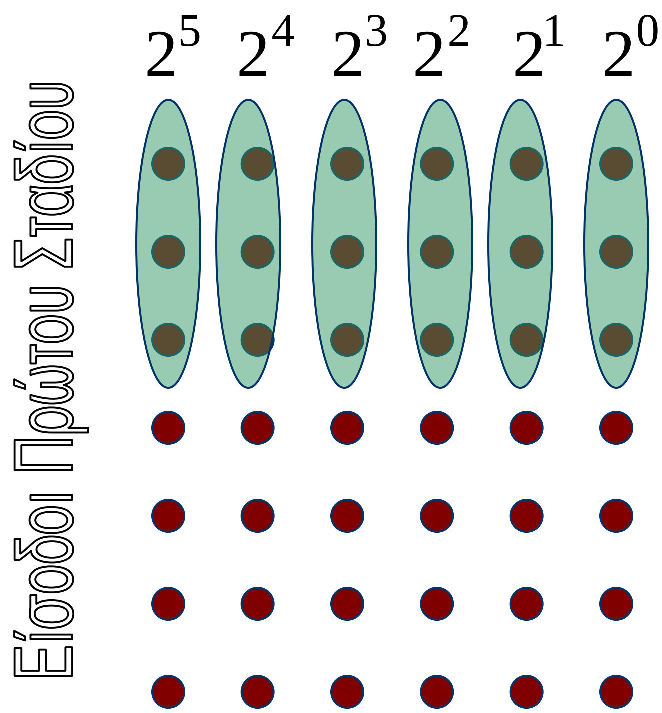
Κατασκευή Δένδρου Wallace

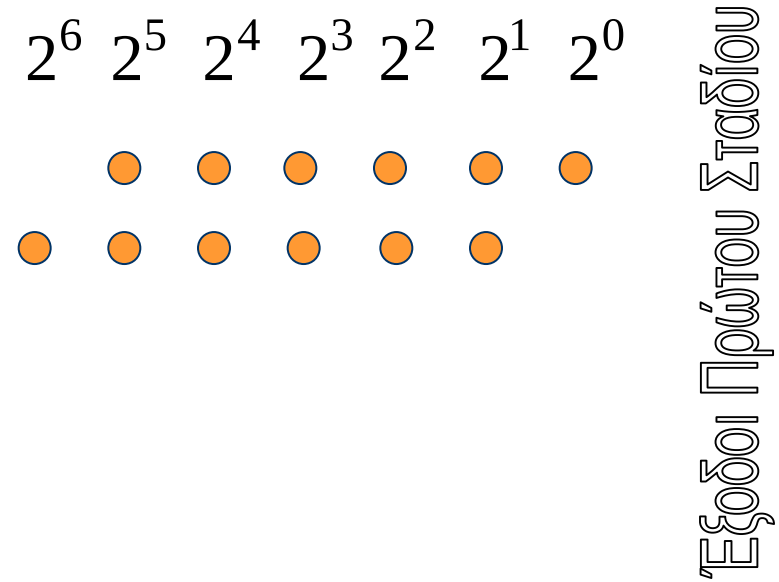
- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.

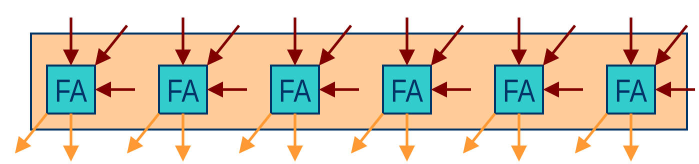


Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.

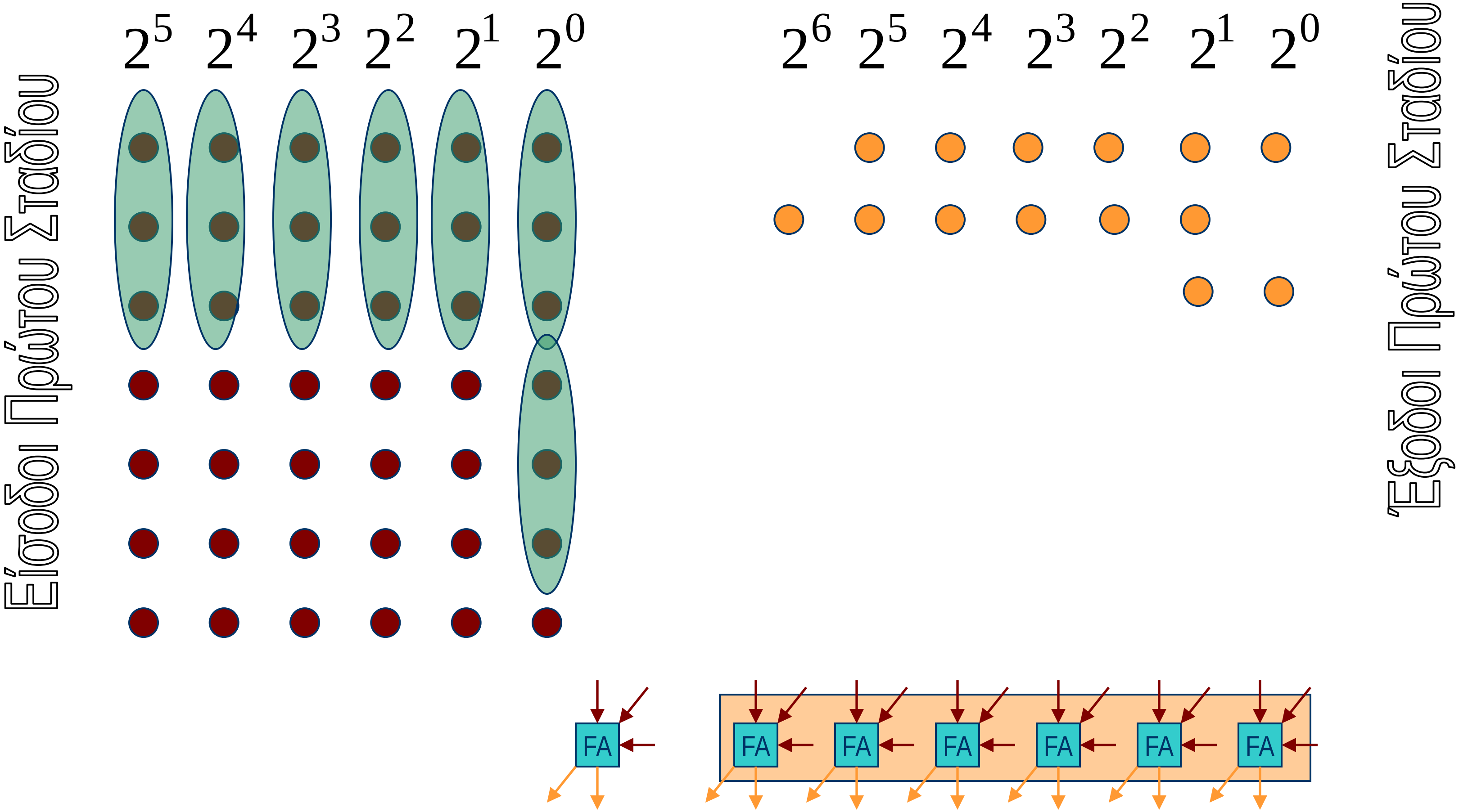






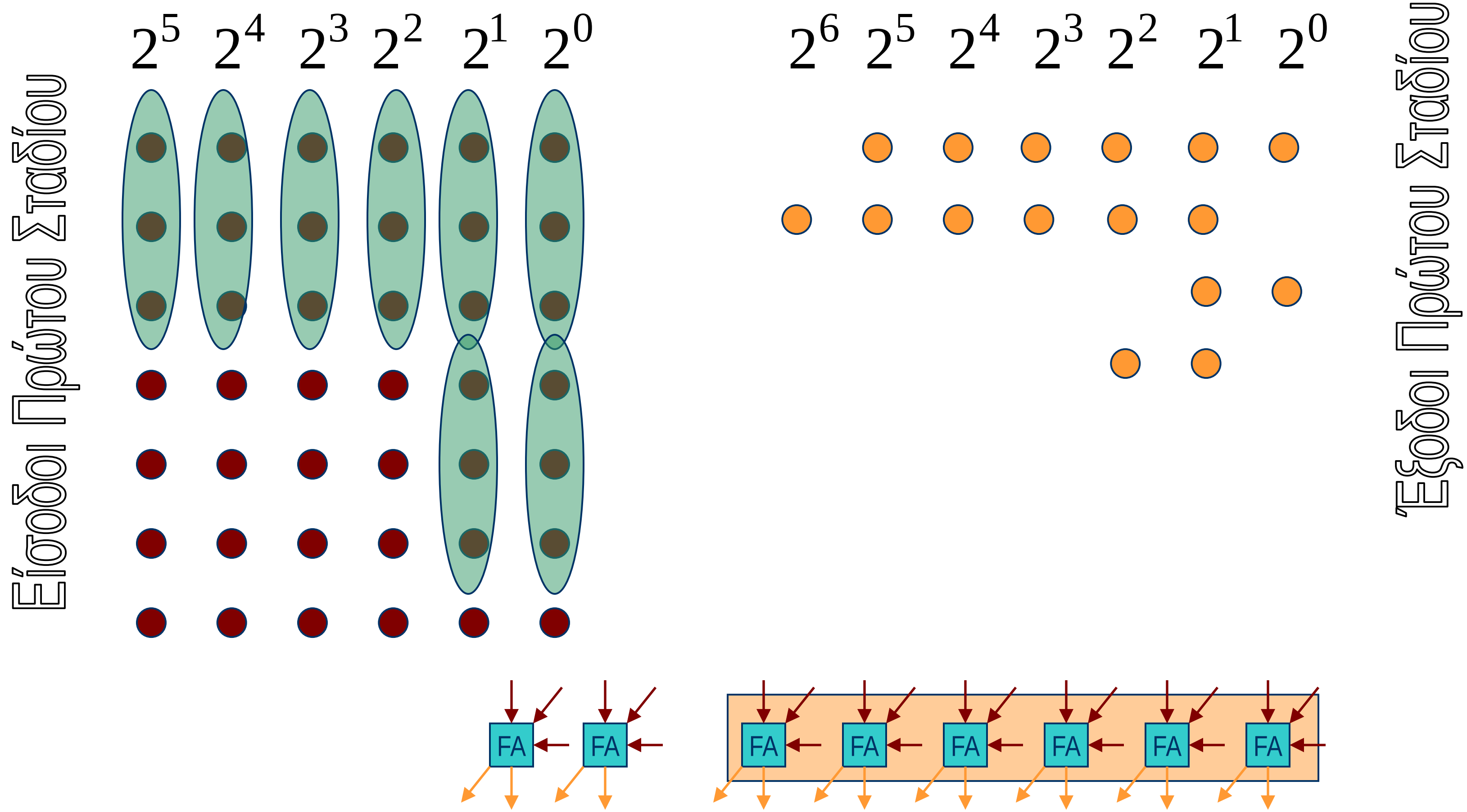
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



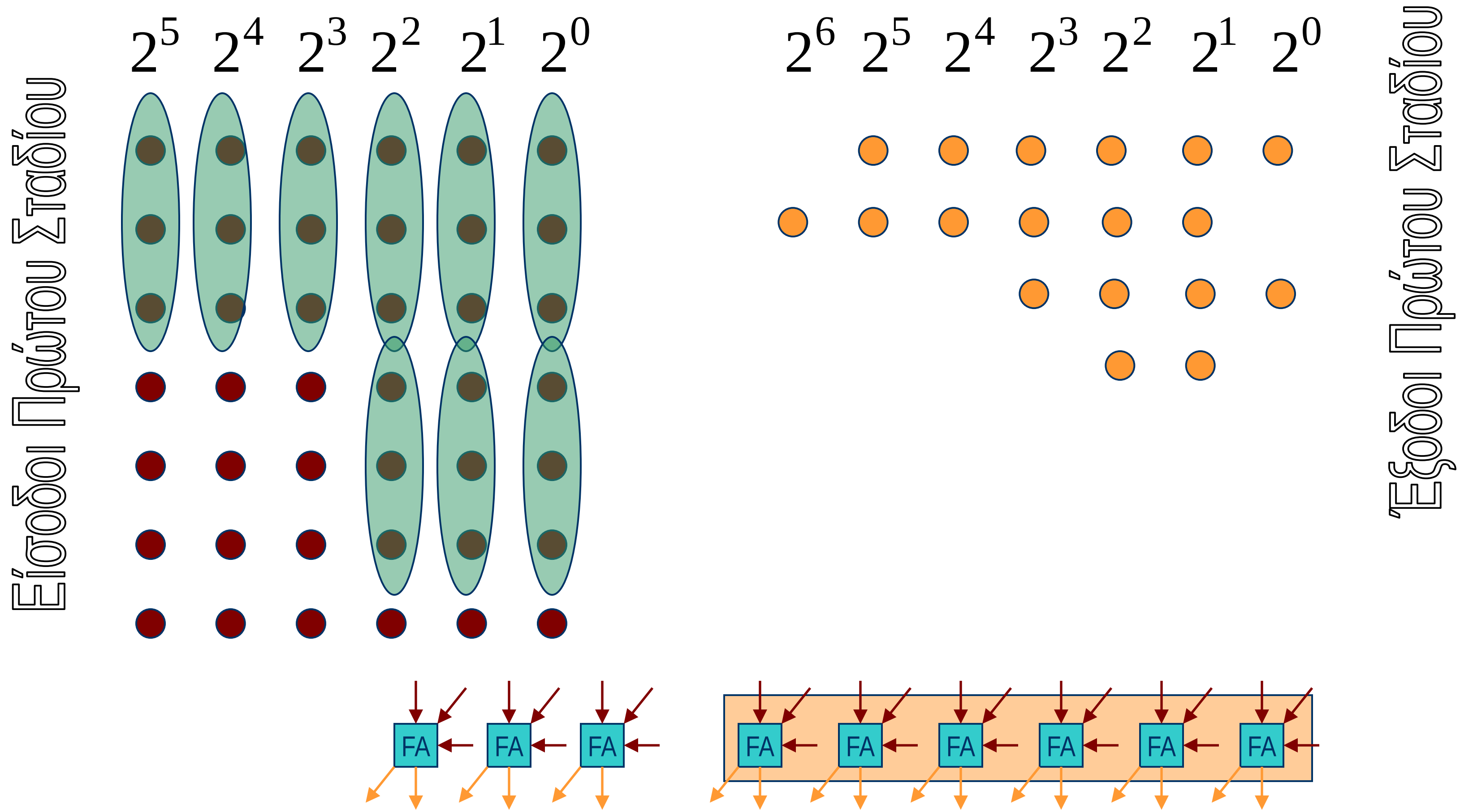
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



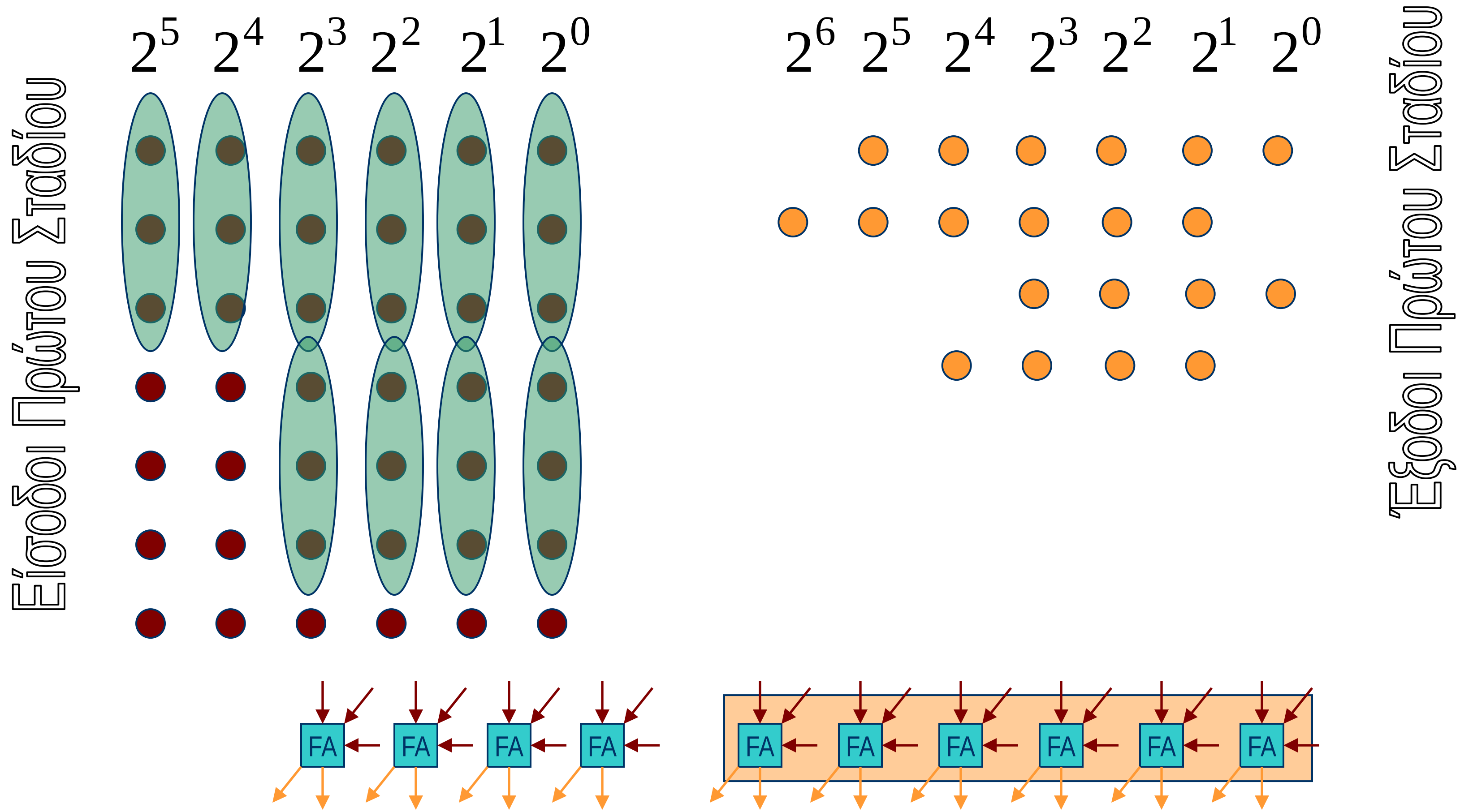
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



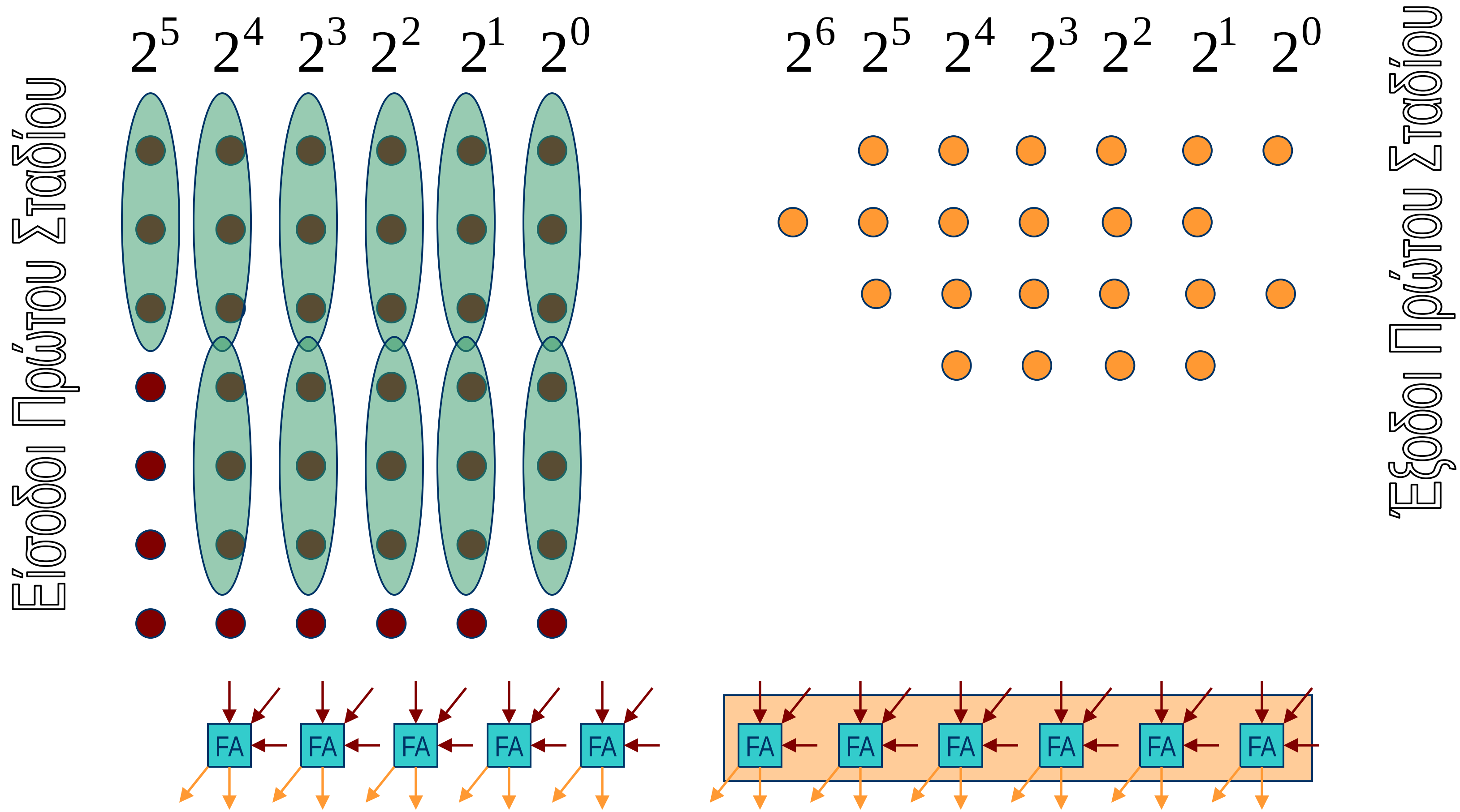
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



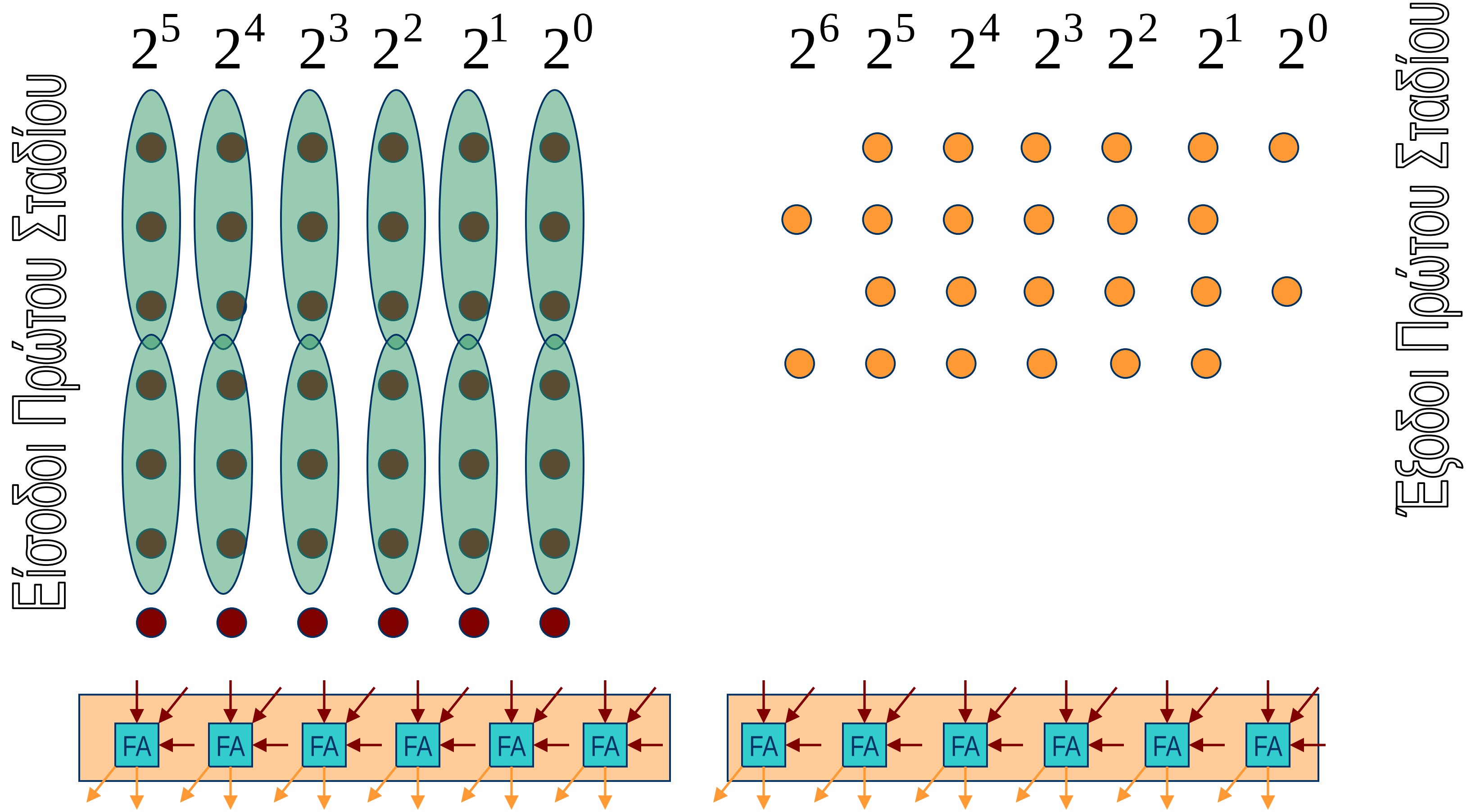
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



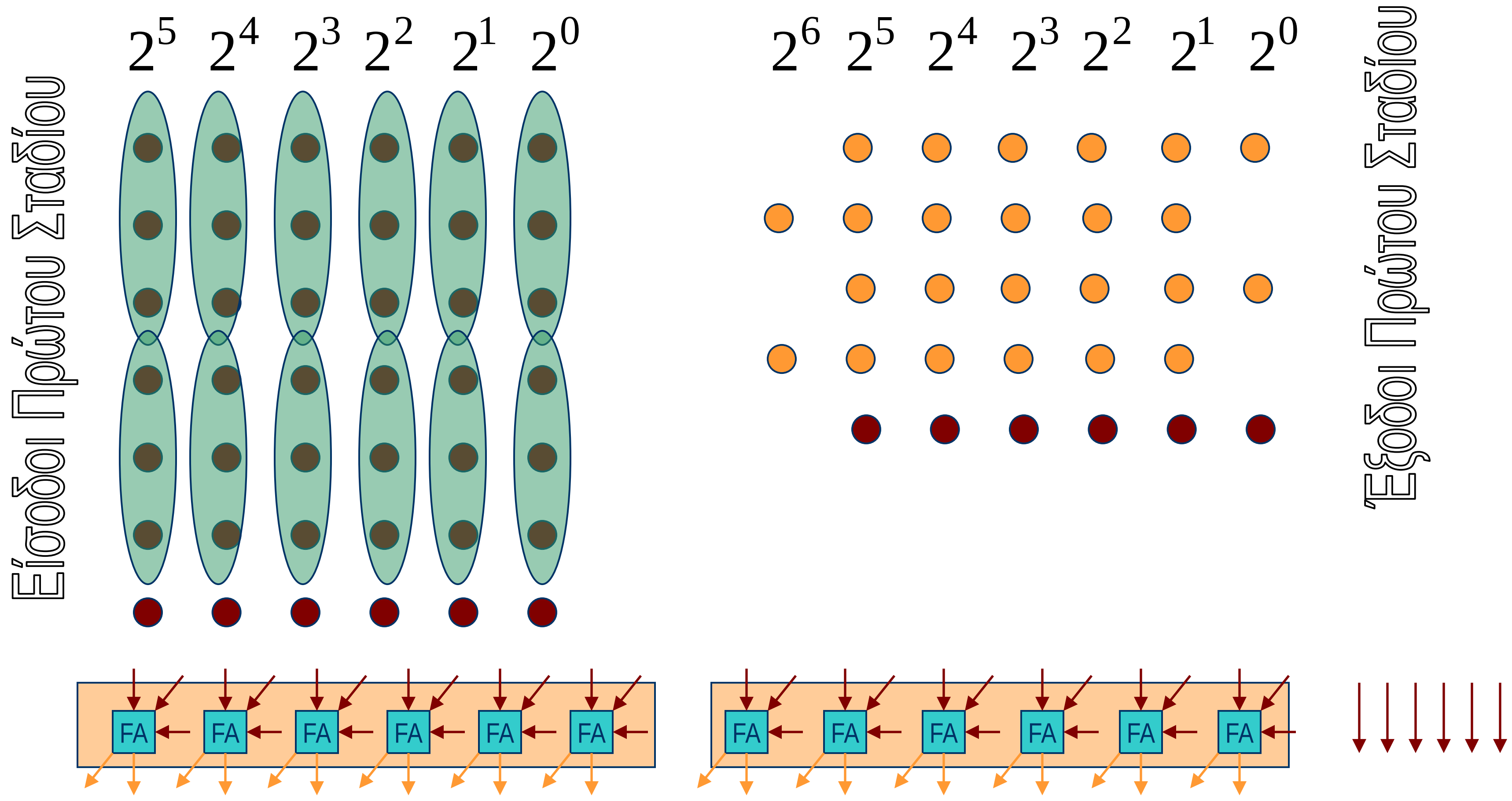
Κατασκευή Δένδρου Wallace

- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



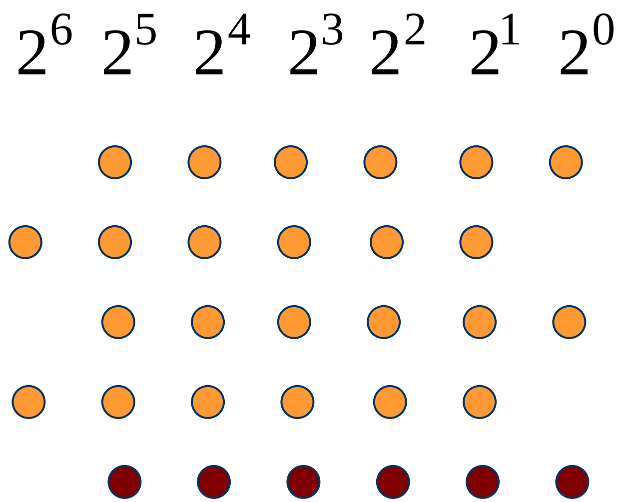
Κατασκευή Δένδρου Wallace

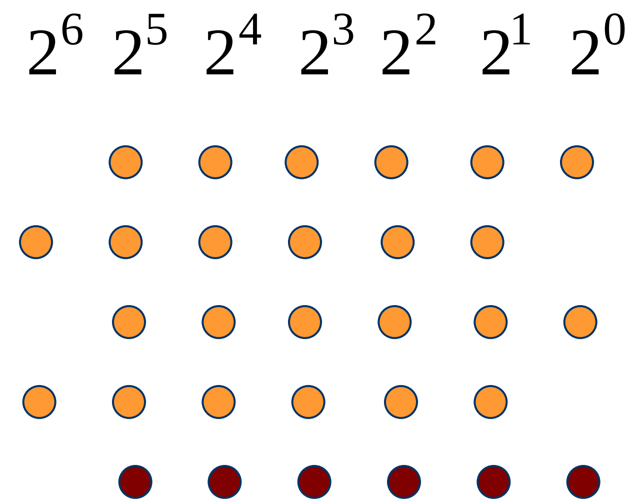
- Άθροιση 7 όρων των 6 bits
  - Για το πρώτο στάδιο: Δύο CSA αθροιστές των 6 bits ο καθένας.
  - Κάθε τελεία, είναι ένα bit της μίας από τις 7 εισόδους.



Κατασκευή Δένδρου Wallace

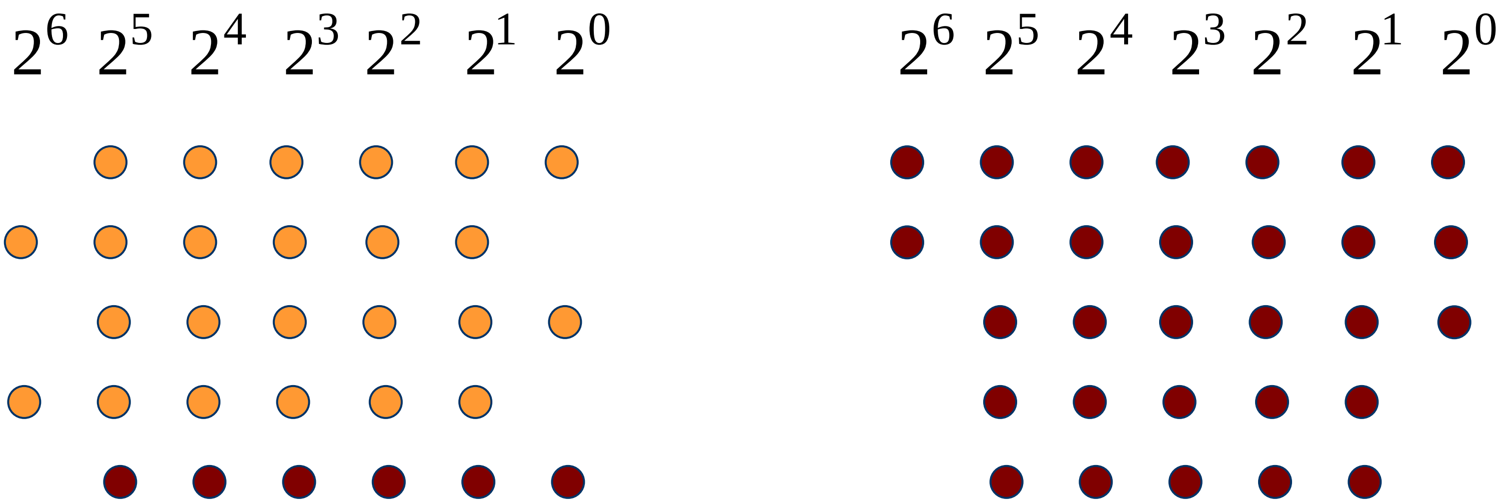
- Είσοδοι στο δεύτερο στάδιο
  - Οι έξοδοι του πρώτου σταδίου γίνονται είσοδοι του δεύτερου, για τους οποίους και επαναλαμβάνουμε τη διαδικασία...





Κατασκευή Δένδρου Wallace

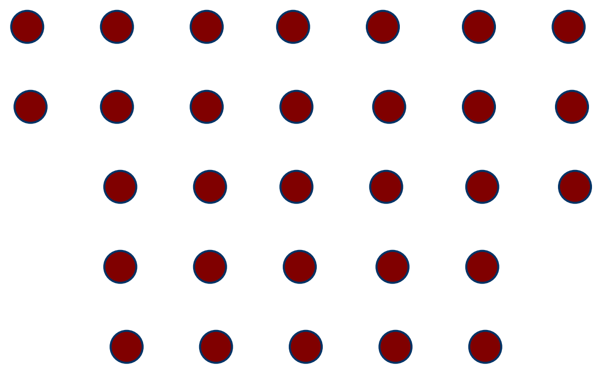
- Είσοδοι στο δεύτερο στάδιο
  - Οι έξοδοι του πρώτου σταδίου γίνονται είσοδοι του δεύτερου, για τους οποίους και επαναλαμβάνουμε τη διαδικασία...



Κατασκευή Δένδρου Wallace

- Παραγωγή Δεύτερου Σταδίου

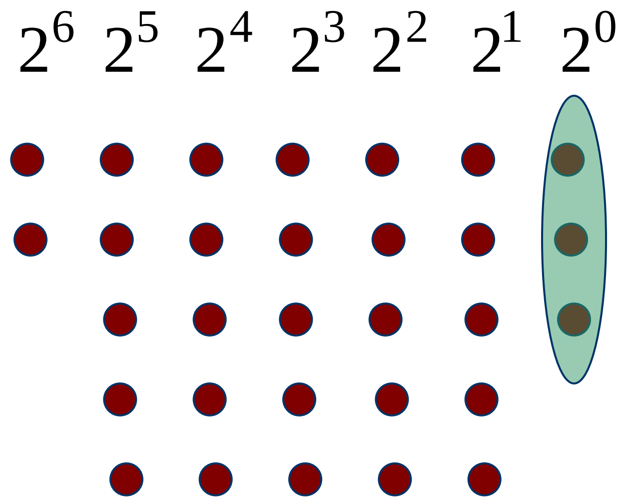
$2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

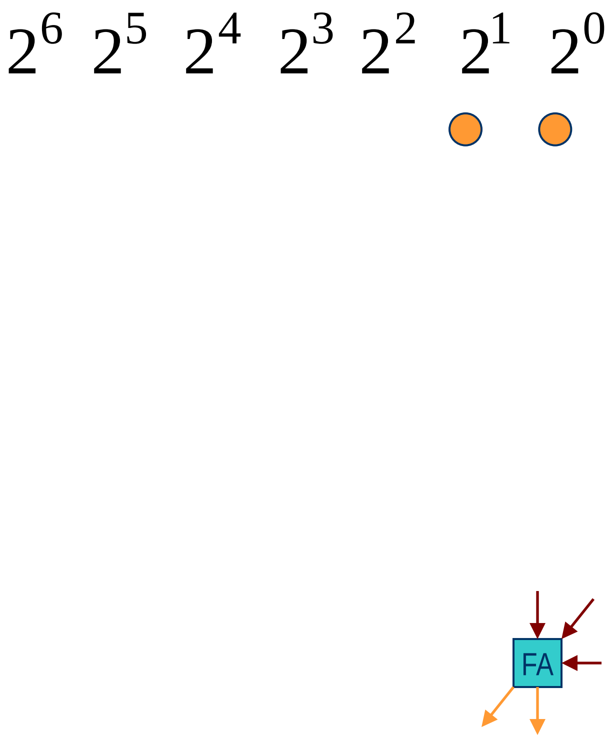


$2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

Κατασκευή Δένδρου Wallace

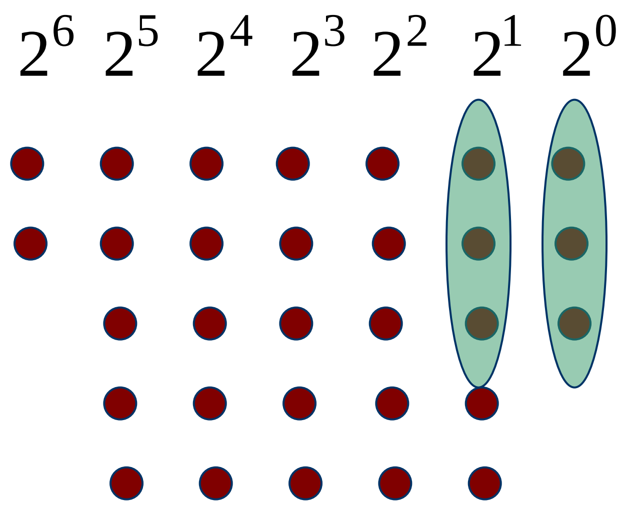
➢ Παραγωγή Δεύτερου Σταδίου

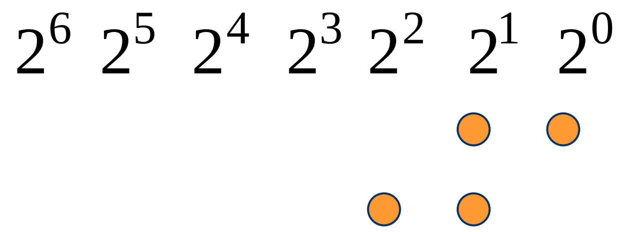


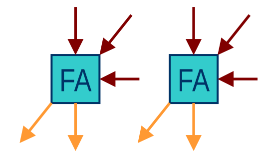


Κατασκευή Δένδρου Wallace

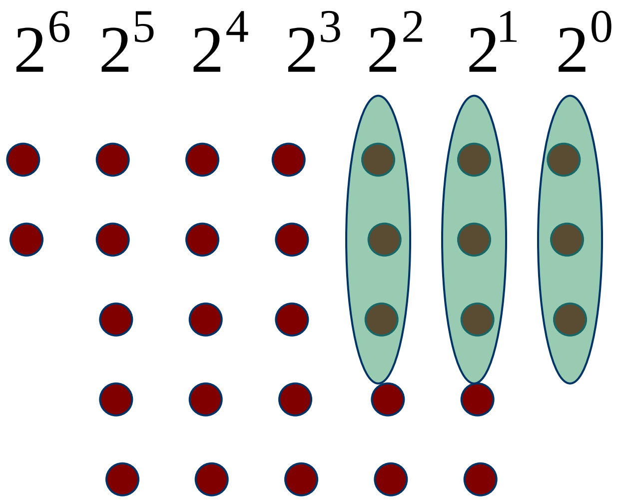
- Παραγωγή Δεύτερου Σταδίου

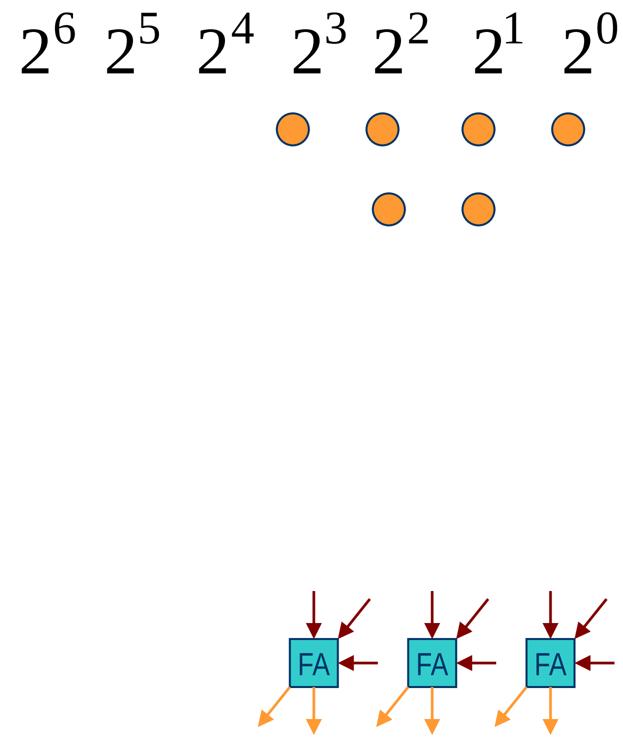






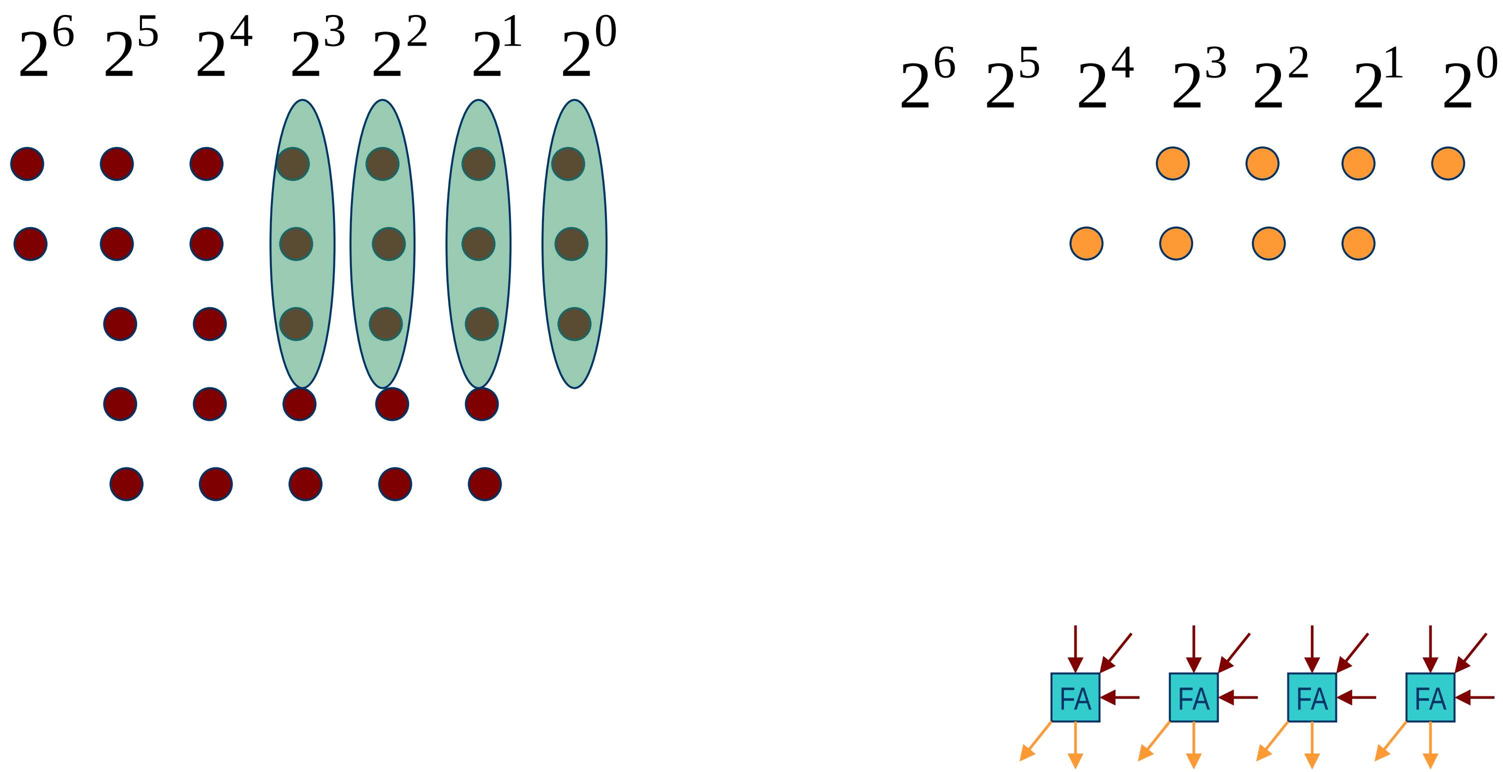
- Παραγωγή Δεύτερου Σταδίου





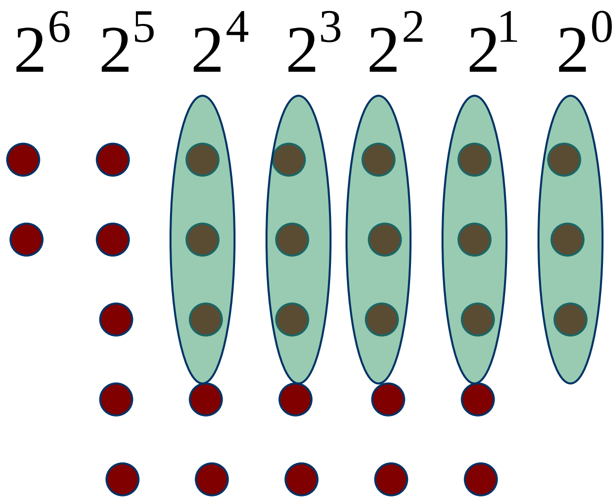
Κατασκευή Δένδρου Wallace

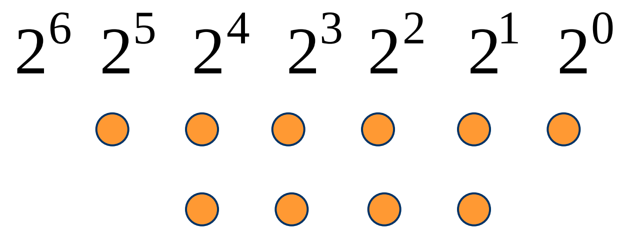
- Παραγωγή Δεύτερου Σταδίου

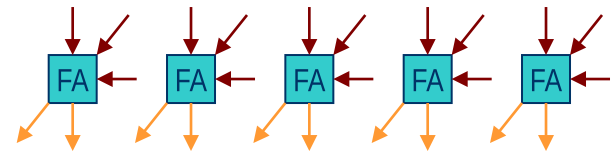


Κατασκευή Δένδρου Wallace

- Παραγωγή Δεύτερου Σταδίου

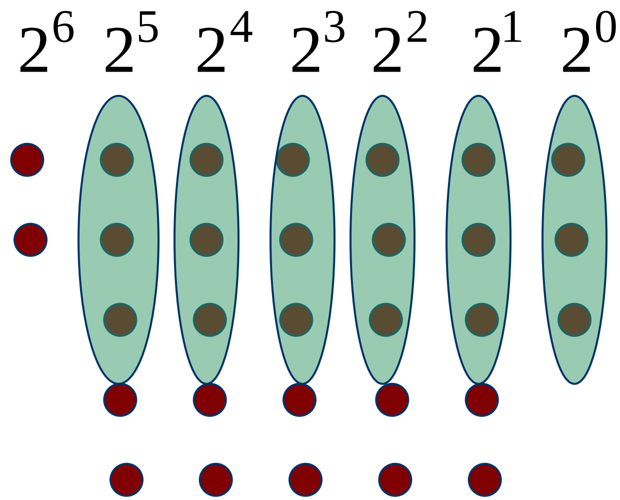


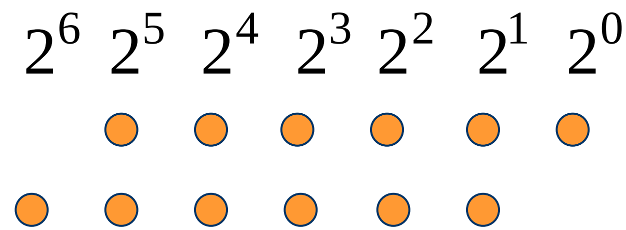


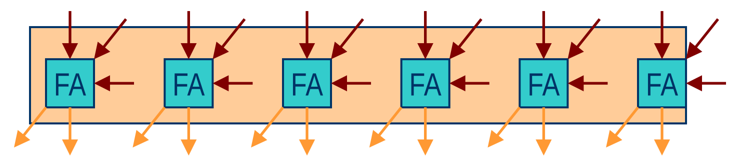


Κατασκευή Δένδρου Wallace

➢ Παραγωγή Δεύτερου Σταδίου

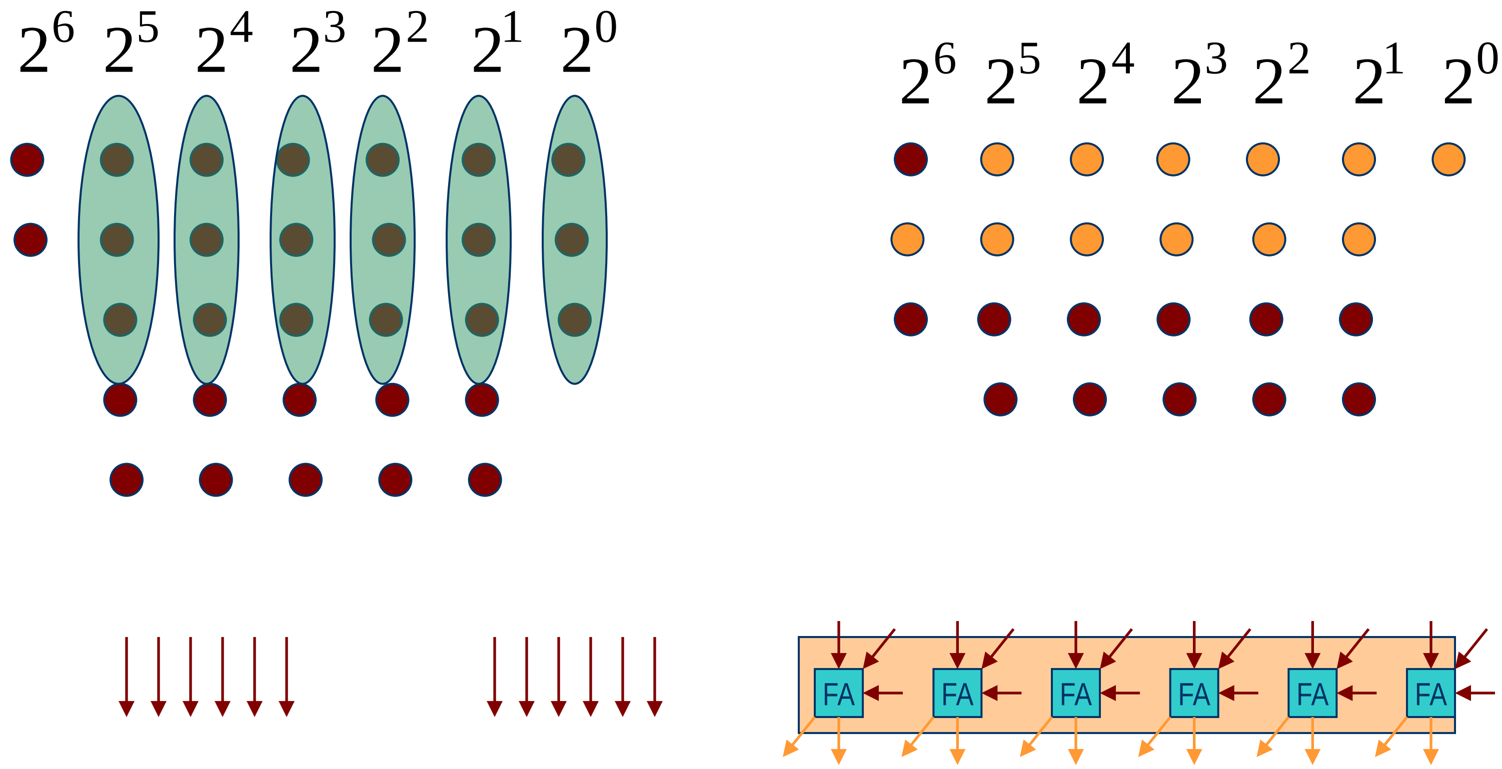






Κατασκευή Δένδρου Wallace

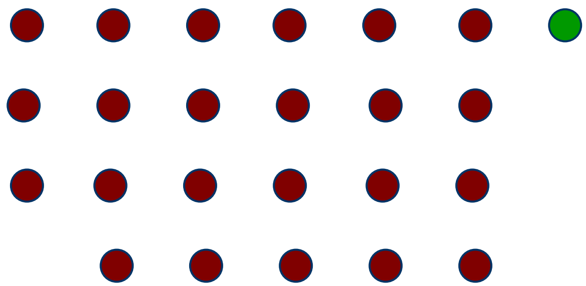
- Παραγωγή Δεύτερου Σταδίου



Κατασκευή Δένδρου Wallace

- Παραγωγή Τρίτου Σταδίου

$2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

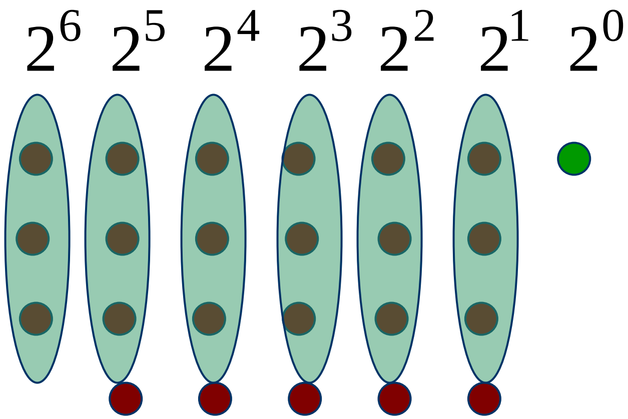


$2^7 \; 2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

● Έτοιμο ψηφίο αποτελέσματος

Κατασκευή Δένδρου Wallace

- Παραγωγή Τρίτου Σταδίου

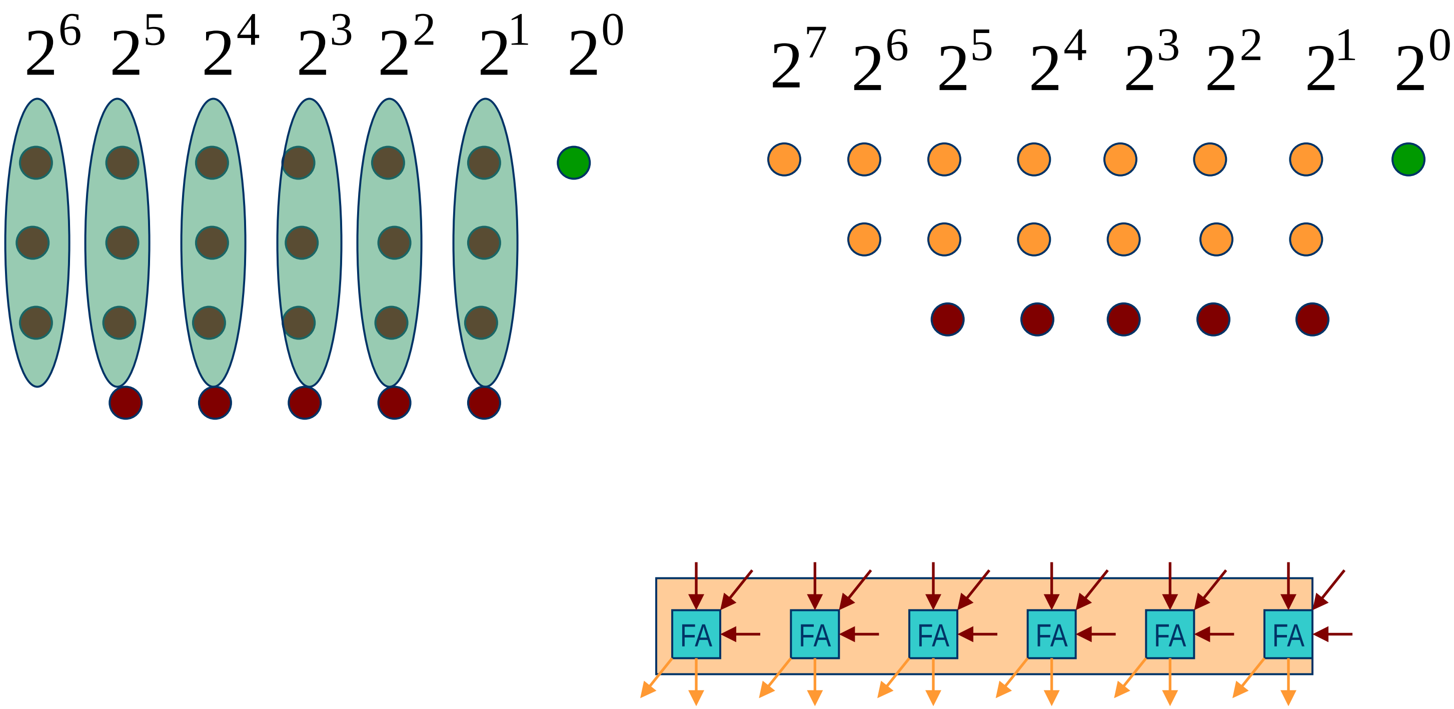


2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

- Έτοιμο ψηφίο αποτελέσματος

Κατασκευή Δένδρου Wallace

- Παραγωγή Τρίτου Σταδίου

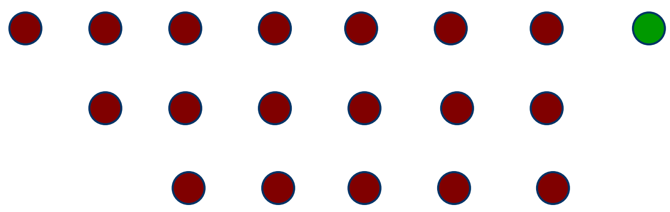


● Έτοιμο ψηφίο αποτελέσματος

Κατασκευή Δένδρου Wallace

- Τέταρτο στάδιο

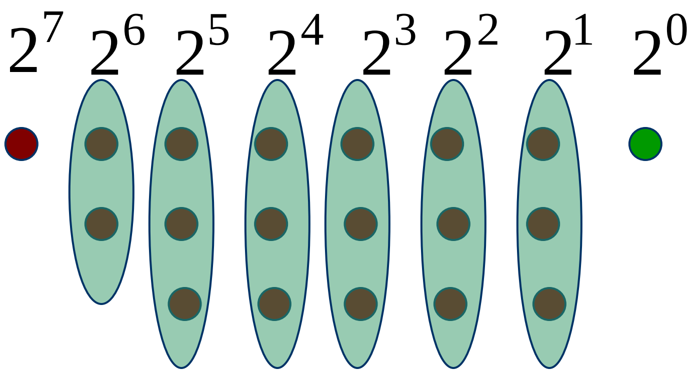
$2^7 \; 2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$



$2^7 \; 2^6 \; 2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0$

Κατασκευή Δένδρου Wallace

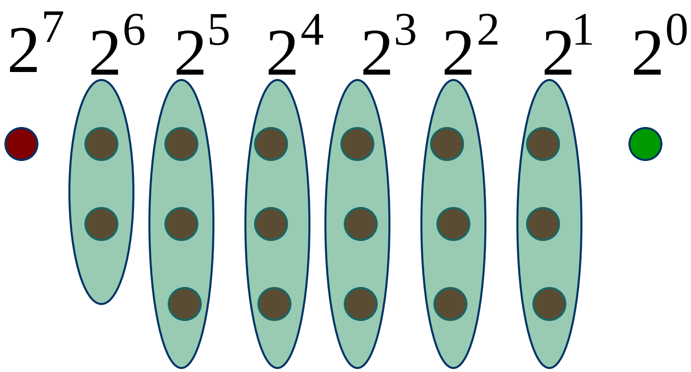
- Τέταρτο στάδιο

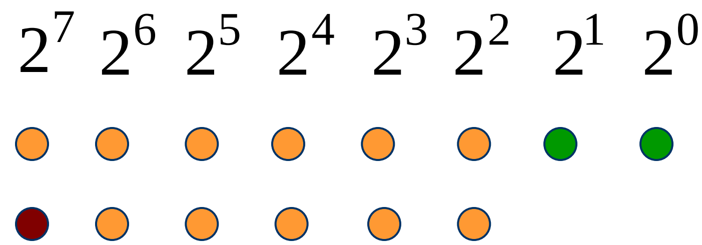


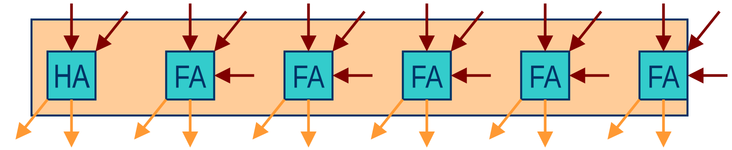
2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

Κατασκευή Δένδρου Wallace

- Τέταρτο στάδιο

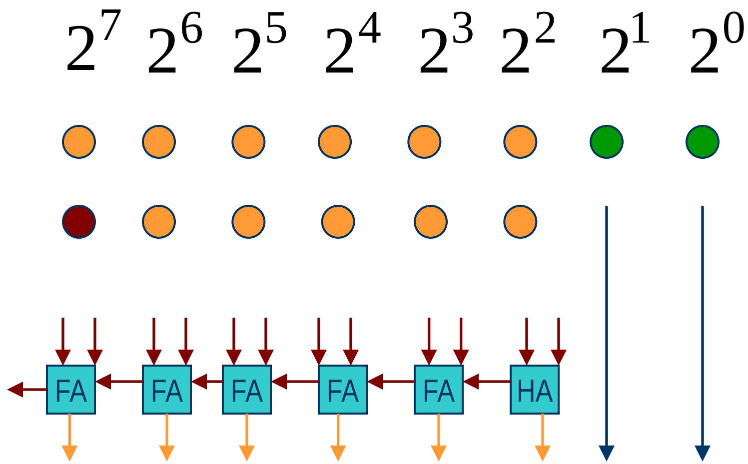




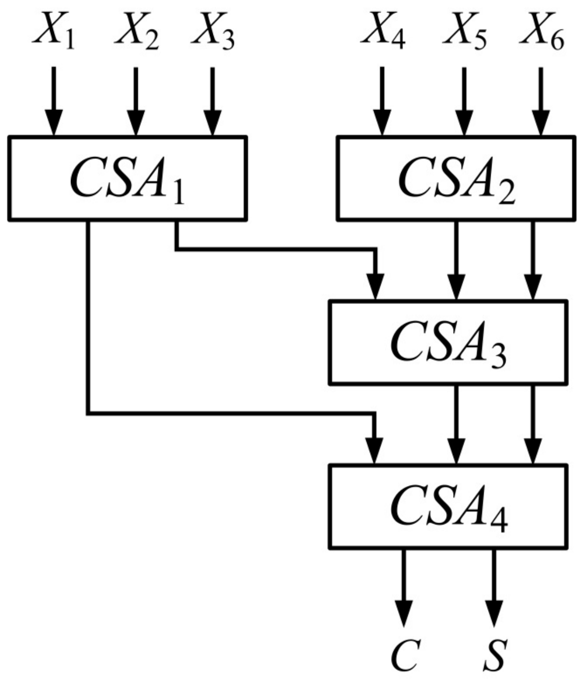


Κατασκευή Δένδρου Wallace

- Carry Propagation
  - Το τελικό στάδιο υλοποιείται με 6-bit carry propagation.



Δένδρα carry save αθροιστών



$$\boxed{h \approx \frac{\log(k/2)}{\log(3/2)} = \log_{3/2}\frac{k}{2}}$$

$$\boxed{T = hT_{CSA} + T_{CPA}}$$

$$\boxed{h(n) = 1 + h\left(\left\lceil \frac{2n}{3} \right\rceil\right)}$$

Αριθμός επιπέδων δένδρων carry save

Πλήθος Όρων *k*	**Αριθμός Επιπέδων, *h***
3	1
4	2
$5 \leq k \leq 6$	3
$7 \leq k \leq 9$	4
$10 \leq k \leq 13$	5
$14 \leq k \leq 19$	6
$20 \leq k \leq 28$	7
$29 \leq k \leq 42$	8
$43 \leq k \leq 63$	9

Βασικό στα δένδρα Dadda

Σημείωμα Χρήσης Έργων Τρίτων

- Τα σχήματα των διαφανειών 35 έως 42 προέρχονται από τις διαφάνειες του συγγράμματος «CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition)», Neil H.E. Weste, David Money Harris, Pearson, 2011.

- Διαθέσιμες στη διαδικτυακή διεύθυνση
  http://pages.hmc.edu/harris/cmosvlsi/4e/index.html
  © 2011 David Money Harris

Σημείωμα Αναφοράς

- Copyright Πανεπιστήμιο Πατρών,
  Βασίλης Παλιουράς, Γεώργιος Θεοδωρίδης,
  «Σχεδιασμός Ολοκληρωμένων Κυκλωμάτων (VLSI) II».
  Έκδοση: 1.0 Πάτρα 2015

- Διαθέσιμο στη διαδικτυακή διεύθυνση
  https://eclass.upatras.gr/courses/EE892/

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου των διδασκόντων καθηγητών.

- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης