

Σχεδίαση Ψηφιακών Κυκλωμάτων

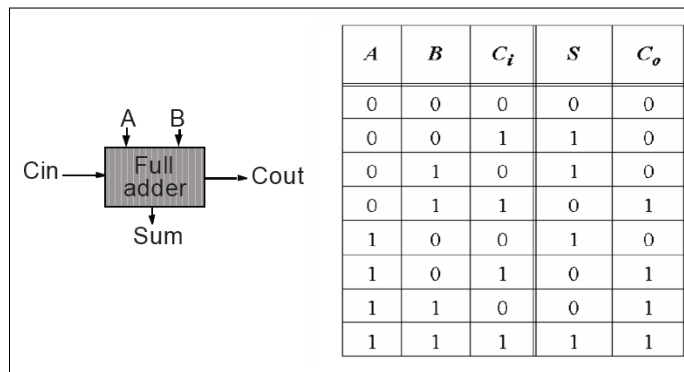
Αριθμητικά κυκλώματα και μνήμες

Νικόλαος Καββαδίας
nkavn@uop.gr

24 Νοεμβρίου 2010

- Ο πλήρης αθροιστής
- Δομές αθροιστών διάδοσης κρατουμένου
- Πολλαπλασιαστές
- Ολισθητές
- Συγκριτές
- Δομές μνήμης ROM
- Κύτταρο μνήμης SRAM

Ο πλήρης αθροιστής (full adder)



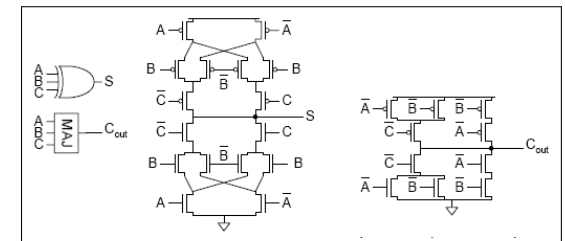
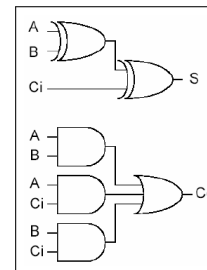
$$S = A \oplus B \oplus C_{in}$$

$$= \overline{A} \overline{B} C_{in} + \overline{A} B \overline{C}_{in} + A \overline{B} \overline{C}_{in} + A B C_{in}$$

$$C_{out} = AB + BC_{in} + AC_{in}$$

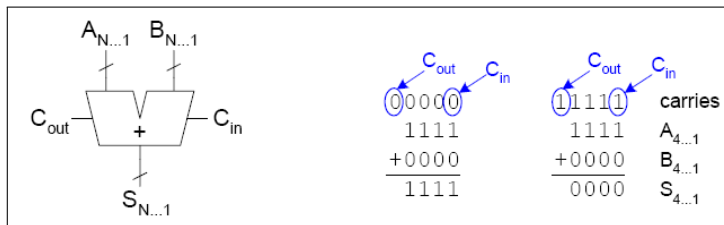
Κυκλωματικός σχεδιασμός του πλήρους αθροιστή

- Το ψηφίο αθροίσματος (S) υπολογίζεται από την XOR τριών εισόδων: $S = A \oplus B \oplus C_{in}$
- Το ψηφίο κρατουμένου (C_{out}) υπολογίζεται από κύκλωμα πλειοψηφίας τριών εισόδων: $C_{out} = MAJ(A, B, C_{in})$



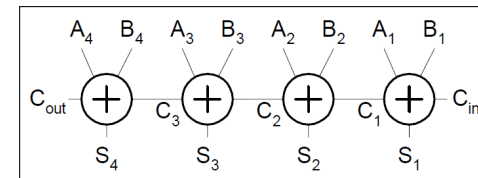
Αθροιστής διάδοσης κρατουμένου (carry propagate adder)

- Carry-propagate adder: CPA
- Κάθε ψηφίο αθροίσματος εξαρτάται από όλα τα προηγούμενα κρατούμενα (από τις προηγούμενες βαθμίδες)
- Χρησιμοποιείται και ως αθροιστής τελικής βαθμίδας σε πολλαπλασιαστές



Αθροιστής ριπής κρατουμένου (RCA: Ripple Carry Adder)

- Η απλούστερη δομή άθροισης δύο δυαδικών αριθμών απρόσμων ή σε συμπλήρωμα-του-2
- Κρίσιμο μονοπάτι από το C_{in} στο C_{out}
- Χρονική απόκριση εξαρτάται από το σχεδιασμό του κυττάρου για τον πλήρη αθροιστή
- Χείριστη καθυστέρηση είναι γραμμική με τον αριθμό των bit: $t_{rca} = (N - 1) \cdot t_{carry} + t_{sum}$ (απόρριψη C_{out})
- Ανίχνευση υπερχείλισης (overflow): $V = C_N \oplus C_{N-1}$
- Πιο σημαντική για την επίδοση του αθροιστή είναι η απόκριση κρατουμένου



Η χρονική καθυστέρηση του αθροιστή ριπής κρατουμένου

- Υποθέτουμε ότι όλες οι πύλες έχουν μοναδιαία καθυστέρηση και ότι οι είσοδοι εφαρμόζονται ταυτοχρόνως κατά το χρονικό σημείο 0
- Για αθροιστή του 1-bit
 - Sum μετά από 2 μονάδες καθυστέρησης πύλης (gate delays)
 - Cout μετά από 2 g.d.
- Για αθροιστή των 2-bit
 - S0 @ 2 C1 @ 2
 - S1 @ 3 C2 @ 4 (worst case: wait for carry)
- Για αθροιστή των 4-bit
 - S0 @ 2 C1 @ 2
 - S1 @ 3 C2 @ 4
 - S2 @ 5 C3 @ 6
 - S3 @ 7 C4 @ 8
- Στη χειρότερη περίπτωση, καθυστέρηση $2N$ g.u. για N-bit RCA

Αθροιστής πρόβλεψης κρατουμένου (CLA: Carry Lookahead Adder)

- Μπορούμε να πετύχουμε καλύτερες επιδόσεις (μικρότερη χρονική καθυστέρηση) αν γράψουμε με διαφορετικό τρόπο τις εξισώσεις υπολογισμού των S_i και C_{i+1}

$$S_i = A_i \oplus B_i \oplus C_i = P_i \oplus C_i$$

$$C_{i+1} = A_i B_i + A_i C_i + B_i C_i = A_i B_i + C_i (A_i + B_i)$$

$$= A_i B_i + C_i (A_i \bar{B}_i + \bar{A}_i B_i + A_i B_i) = A_i B_i + C_i (A_i \oplus B_i)$$

$$= G_i + P_i C_i$$

όπου: $P_i = A_i \oplus B_i$ (propagate) και $G_i = A_i \cdot B_i$ (generate)

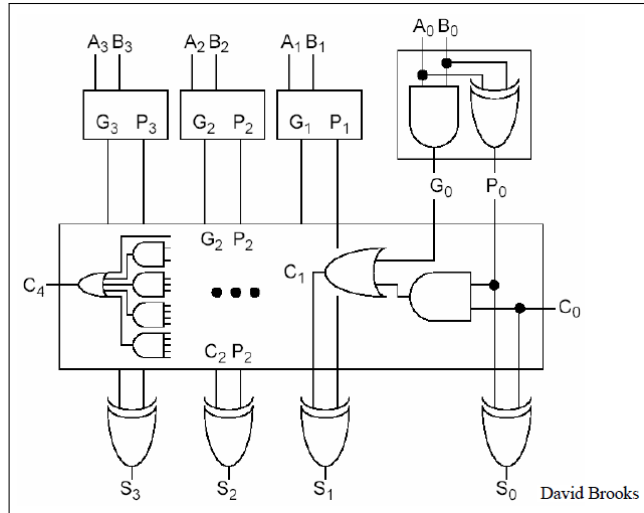
- Έτσι έχουμε τις εξής προβλέψεις κρατουμένων:

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

- Το κρατούμενο C_i αποτελεί το άθροισμα $i + 1$ παραγόντων και διαθέτει τουλάχιστον έναν παράγοντα $i + 1$ όρων

Σχηματικό διάγραμμα CLA των 4-bit



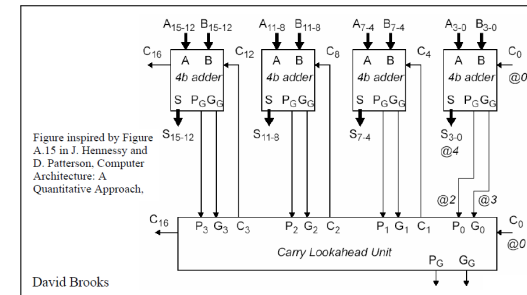
Νικόλαος Καββαδίας nkavn@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Ιεραρχική προσέγγιση για το σχεδιασμό CLA περισσότερων ψηφίων

- Στο 1ο επίπεδο (χαμηλότερο) χρησιμοποιούνται αθροιστές πρόβλεψης κρατούμενου των 4-bit
- Κάθε αθροιστής των 4-bit υπολογίζει τα δικά του ψηφία πρόβλεψης ομάδας P_G και G_G :

$$P_G = P_3 \cdot P_2 \cdot P_1 \cdot P_0$$

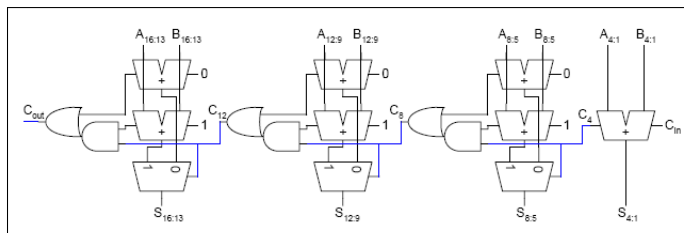
$$G_G = G_3 + G_2 P_3 + G_1 P_3 P_2 + G_0 P_3 P_2 P_1$$



Νικόλαος Καββαδίας nkavn@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Αθροιστής επιλογής κρατούμενου (carry-select adder)

- Ο αθροιστής επιλογής κρατούμενου προϋπολογίζει και τις δύο πιθανές εξόδους για το κρατούμενο εισόδου X (δηλαδή για $X = 0$ και $X = 1$) για ομάδες των n bit
- Το X αποτελεί κρατούμενο εξόδου από την προηγούμενη βαθμίδα διάδοσης κρατούμενου



Νικόλαος Καββαδίας nkavn@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Υπολογιστική πολυπλοκότητα των βασικών τοπολογιών άθροισης δύο ορισμάτων

- Από τις τρεις βασικές τοπολογίες αθροιστών: RCA, CLA, CSA, τα θεωρητικά καλύτερα χαρακτηριστικά διαθέτει ο CLA (αθροιστής πρόβλεψης κρατούμενου)
- Ο RCA έχει τη χειρότερη επίδοση και ο CSA έχει ενδιάμεσες επιδόσεις
- Στις διατάξεις FPGA, τα διαθέσιμα κελιά είναι βελτιστοποιήσιμα για τοπολογίες ριπής (fast carry chain). Στην περίπτωση αυτή, μπορεί να είναι συμφέρουσα ακόμη και η τοπολογία RCA

| Τοπολογία | Χρονική πολυπλοκότητα | Χωρική πολυπλοκότητα |
|-----------|-----------------------|----------------------|
| RCA | $O(n)$ | $O(n)$ |
| CLA | $O(\log(n))$ | $O(n \cdot \log(n))$ |
| CSA | $O(\sqrt{n})$ | $O(n)$ |

Νικόλαος Καββαδίας nkavn@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Η πράξη του πολλαπλασιασμού

- Πολλαπλασιασμός δύο δυαδικών αριθμών των M και N bit αντίστοιχα
- Η διαδικασία (‘σχολική’ μέθοδος ή “pencil-and-paper”) παράγει N μερικά γινόμενα των M -bit
- Η άθροιση των μερικών γινομένων δίνει το τελικό γινόμενο που έχει εύρος $(M + N)$ -bit
- Παράδειγμα:

| | | | |
|---------------|---|-------------|------------------------------------|
| 1100 | : | $(12)_{10}$ | πολλαπλασιαστέος (multiplicand) |
| \times 0101 | : | $(5)_{10}$ | πολλαπλασιαστής (multiplier) |
| 1100 | | | μερικά γινόμενα (partial products) |
| 0000 | | | |
| 1100 | | | |
| 0000 | | | |
| 00111100 | : | $(60)_{10}$ | γινόμενο (product) |

Γενική μορφή του πολλαπλασιασμού

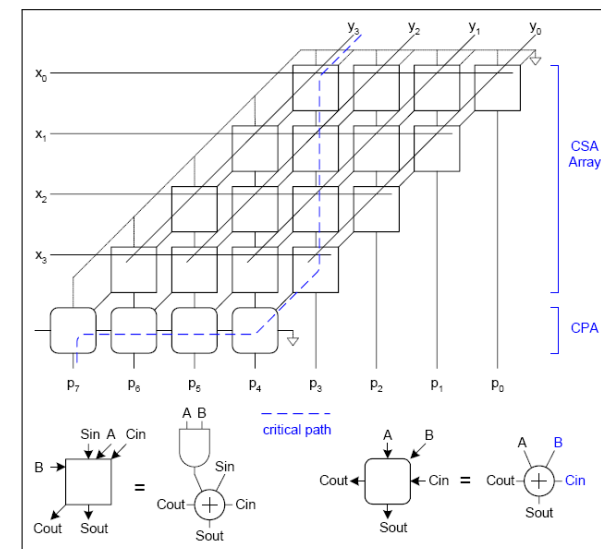
- Πολλαπλασιαστέος: $Y = (y_{M-1}, y_{M-2}, \dots, y_1, y_0)$
- Πολλαπλασιαστής: $X = (x_{N-1}, x_{N-2}, \dots, x_1, x_0)$
- Γινόμενο: $P = (\sum_{j=0}^{M-1} y_j 2^j) \cdot (\sum_{i=0}^{N-1} x_i 2^i) = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} x_i y_j 2^{i+j}$
- Πολλαπλασιασμός 2 αριθμών των 6-bit:

| | | | | | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-------|-------|
| | | | | y_5 | y_4 | y_3 | y_2 | y_1 | y_0 | | |
| | | | | x_5 | x_4 | x_3 | x_2 | x_1 | x_0 | | |
| | | | | $x_0 y_5$ | $x_0 y_4$ | $x_0 y_3$ | $x_0 y_2$ | $x_0 y_1$ | $x_0 y_0$ | | |
| | | | $x_1 y_5$ | $x_1 y_4$ | $x_1 y_3$ | $x_1 y_2$ | $x_1 y_1$ | $x_1 y_0$ | | | |
| | | $x_2 y_5$ | $x_2 y_4$ | $x_2 y_3$ | $x_2 y_2$ | $x_2 y_1$ | $x_2 y_0$ | | | | |
| | $x_3 y_5$ | $x_3 y_4$ | $x_3 y_3$ | $x_3 y_2$ | $x_3 y_1$ | $x_3 y_0$ | | | | | |
| $x_4 y_5$ | $x_4 y_4$ | $x_4 y_3$ | $x_4 y_2$ | $x_4 y_1$ | $x_4 y_0$ | | | | | | |
| $x_5 y_5$ | $x_5 y_4$ | $x_5 y_3$ | $x_5 y_2$ | $x_5 y_1$ | $x_5 y_0$ | | | | | | |
| p_{11} | p_{10} | p_9 | p_8 | p_7 | p_6 | p_5 | p_4 | p_3 | p_2 | p_1 | p_0 |

Ο πολλαπλασιαστής τύπου πίνακα (array multiplier)

- Η απλούστερη κυκλωματική δομή πολλαπλασιασμού
- Αποτελείται από μία ορθογώνια διάταξη κελιών για τον υπολογισμό και τη διάδοση των μερικών γινομένων
- Κάθε κελί αποτελείται από έναν πλήρη αθροιστή (πρώτη σειρά) ή μία πύλη AND (σήμα generate) και έναν πλήρη αθροιστή
- Η διάταξη αυτή υλοποιείται με αθροιστές αποθήκευσης κρατουμένου
- Η τελική άθροιση πραγματοποιείται από έναν αθροιστή διάδοσης κρατουμένου

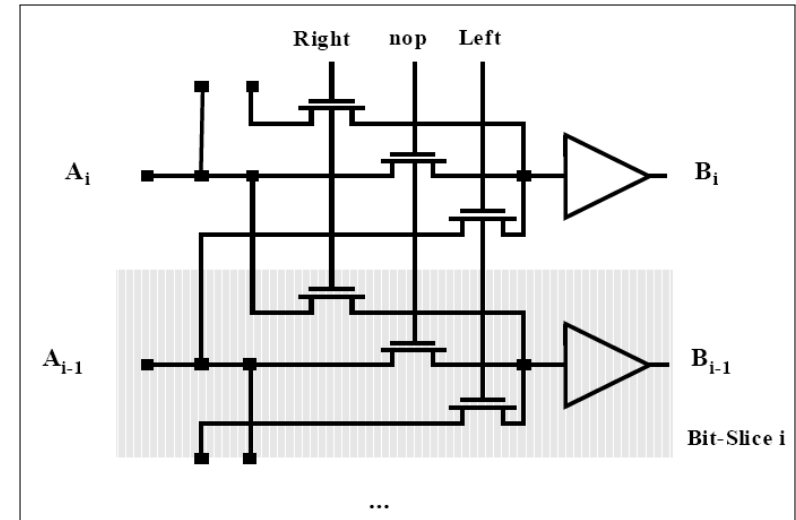
Κυκλωματική σχεδίαση του πολλαπλασιαστή τύπου πίνακα



Ολισθητές (shifters)

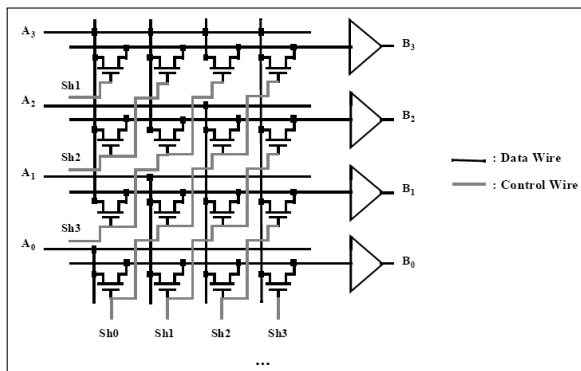
- Πράξεις ολίσθησης (shift) και περιστροφής (rotation)
- Ολίσθηση
 - Λογική ολίσθηση: μετακίνηση των ψηφίων κατά i θέσεις και συμπλήρωση των κενών θέσεων με 0
 - Λογική ολίσθηση προς τα αριστερά (LSL: Logical Shift Left)
 - Παράδειγμα: $1011 \text{ LSL } 1 = 0110$
 - Λογική ολίσθηση προς τα δεξιά (LSR: Logical Shift Right)
 - Παράδειγμα: $1011 \text{ LSR } 1 = 0101$
 - Αριθμητική ολίσθηση: μετακίνηση των ψηφίων κατά i θέσεις και συμπλήρωση με το ψηφίο προσήμου (sign bit)
 - Αριθμητική ολίσθηση προς τα δεξιά (ASR: Arithmetic Shift Right)
 - Παράδειγμα: $1011 \text{ ASR } 1 = 1101$
- Περιστροφή
 - Δεξιά (ROTR) και αριστερή (ROTL)
 - Παραδείγματα: $1011 \text{ ROTR } 1 = 1101$, $1011 \text{ ROTL } 1 = 0111$

Διαδικός ολισθητής δεξιά/αριστερά



Βαρελοειδής ολισθητής (barrel shifter)

- Υλοποιεί όλα τα είδη ολίσθησης και περιστροφής
- Απαιτεί μεγάλη επιφάνεια υλικού λόγω των διασυνδέσεών του
- Βαρελοειδής ολισθητής των 4-bit

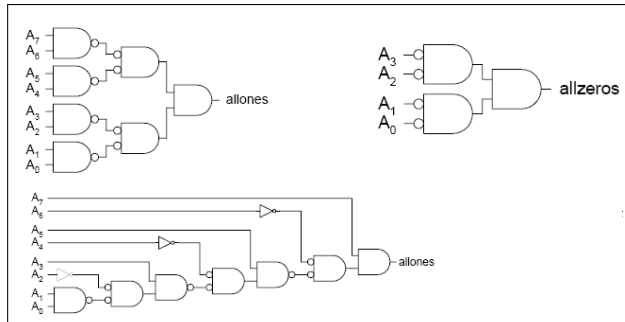


Συγκριτές (comparators)

- Ανιχνευτής μηδενός (zero's detector)
- Ανιχνευτής μονάδων (one's detector)
- Συγκριτής ισότητας (equality comparator):
 $A == B$, $A != B$
- Συγκριτής μεγέθους (magnitude comparator):
 $A > B$, $A >= B$, $A < B$, $A <= B$

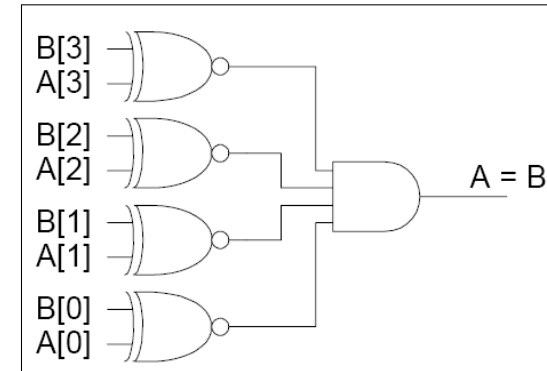
Ανιχνευτές μηδενός και μονάδων

- Ανιχνευτής N μονάδων: πύλη AND των N εισόδων (υλοποιημένη από ισοσταθμισμένο δένδρο πυλών)
- Ανιχνευτής μηδενός: με ανίχνευση μονάδων στο συμπλήρωμα-ως-προς 1 ή με πύλη NOR των N εισόδων
- Κυκλώματα



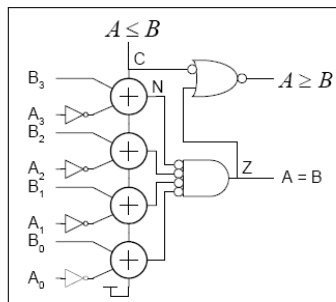
Συγκριτής ισότητας

- Σε επίπεδο bit, με πύλη XNOR των 2 εισόδων (τελεστής ισοδυναμίας)
- Για λέξη των N bit τα αποτελέσματα των πυλών XNOR οδηγούν πύλη AND των N εισόδων

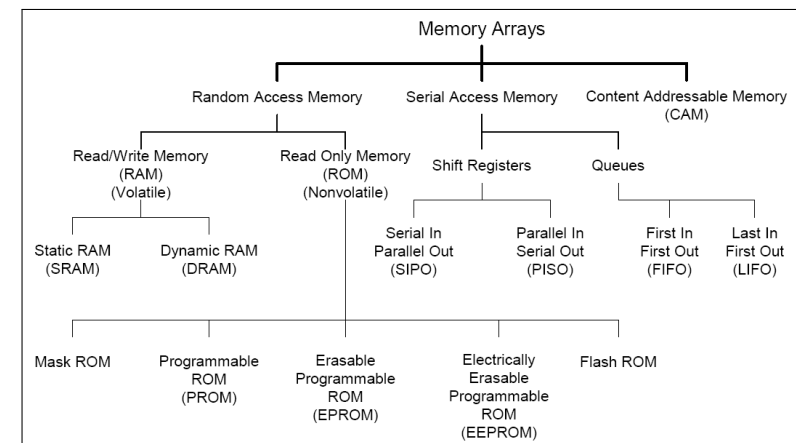


Συγκριτής ανισότητας

- Υπολογίζουμε τη διαφορά $B - A$ και παρατηρούμε το πρόσημο του αποτελέσματος της αφαίρεσης
- Η αφαίρεση δύο αριθμών σε συμπλήρωμα-του-2 γίνεται ως εξής: $B - A = B + \bar{A} + 1$
- Για απρόσημους αριθμούς, παρατηρούμε το κρατούμενο εξόδου

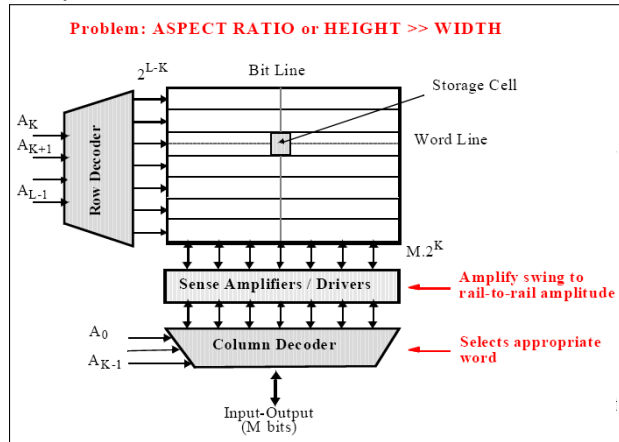


Κατηγοριοποίηση των συστοιχιών μνήμης (memory arrays)



Βασική οργάνωση μιας μνήμης

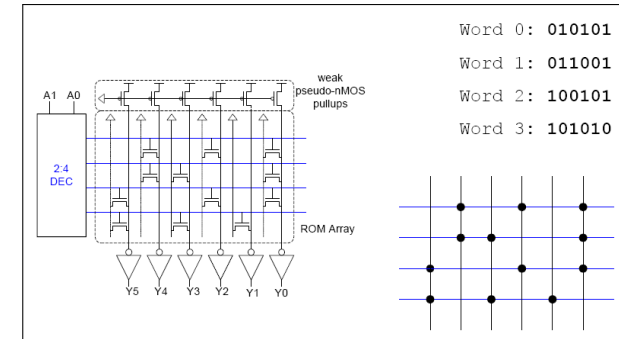
- Δομικά στοιχεία μιας τυπικής μνήμης: αποκωδικοποιητής σειράς, αποκωδικοποιητής στήλης, ενισχυτές αίσθησης (sense amplifiers), κελιά αποθήκευσης



Νικόλαος Καββαδίας nkavv@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

ROM (Μνήμη μόνο ανάγνωσης)

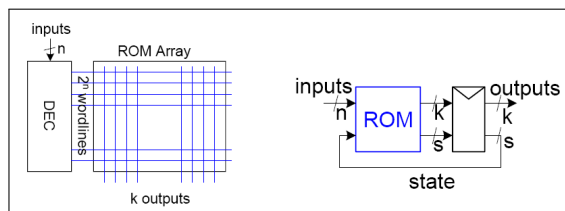
- Διατηρεί τα περιεχόμενά της και μετά το σβήσιμο της τροφοδοσίας
- Κάθε bit κωδικοποιείται με τη βοήθεια ενός τρανζίστορ
- Παράδειγμα: ROM 4 λέξεων των 6-bit



Νικόλαος Καββαδίας nkavv@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Υλοποίηση λογικής με μνήμες ROM

- Η ROM μπορεί να χρησιμοποιηθεί ως πίνακας αναζήτησης (LUT) για την υλοποίηση λογικών συναρτήσεων (δηλαδή των πινάκων αληθείας τους)
- Για n εισόδους και k εξόδους απαιτούνται 2^n k -bit λέξεις
- FSM: n είσοδοι, k έξοδοι, και s ψηφία για την κωδικοποίηση της κατάστασης
- Υλοποίηση με $2^{n+s}(k+s)$ bit ROM και $(k+s)$ bit καταχωρητή



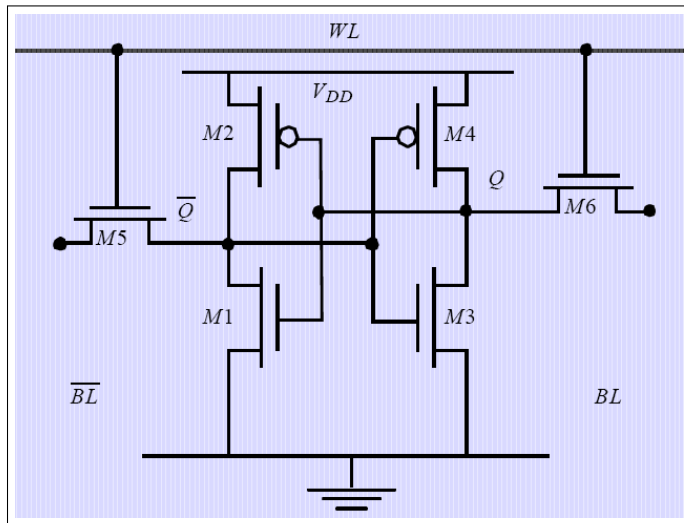
Νικόλαος Καββαδίας nkavv@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Μνήμη RAM

- RAM: Random Access Memory: Μνήμη Τυχαίας Προσπέλασης
- Λειτουργίες ανάγνωσης και εγγραφής
- Βασικοί τύποι RAM: SRAM και DRAM
- SRAM: Static RAM (στατική RAM)
 - Τα δεδομένα παραμένουν αποθηκευμένα όσο εφαρμόζεται τάση τροφοδοσίας
 - Κύτταρο μνήμης των 6 τρανζίστορ (6T)
 - Απλή οργάνωση
- DRAM: Dynamic RAM (δυναμική RAM)
 - Απαιτείται περιοδική ανανέωση των περιεχομένων της (refresh)
 - Κύτταρο μνήμης των 1-3 τρανζίστορ
 - Πιο αργή

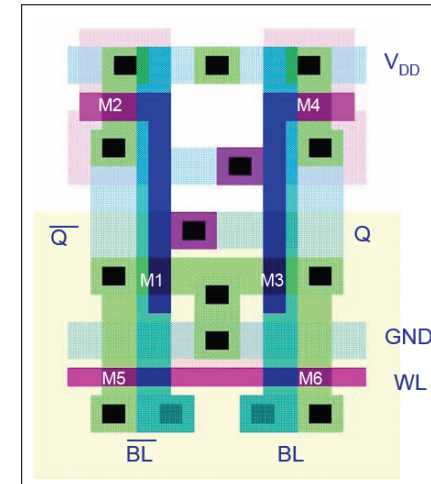
Νικόλαος Καββαδίας nkavv@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Τυπικό κελί μνήμης SRAM 6T σε τεχνολογία CMOS



Νικόλαος Καββαδίας nkann@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

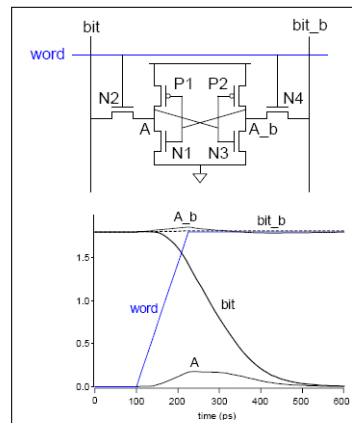
Φυσική σχεδίαση του κελιού μνήμης 6T



Νικόλαος Καββαδίας nkann@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

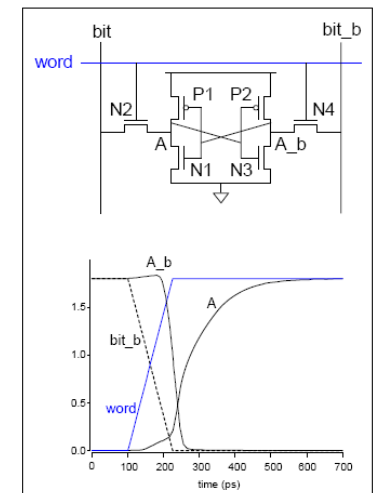
Διαδικασία ανάγνωσης (SRAM read)

- Προφόρτιση των γραμμών bit και bit_b σε υψηλή στάθμη
- Ενεργοποίηση της γραμμής word
- Μία από τις δύο γραμμές (bit ή bit_b) θα οδηγηθεί σε χαμηλή στάθμη
- Έστω $A = 0$, $A_b = 1$
- Αποφόρτιση της bit, bit_b σε υψηλή στάθμη
- Για ευσταθή ανάγνωση, ο A δεν πρέπει να μεταβληθεί



Διαδικασία εγγραφής (SRAM write)

- Οδήγηση της μία γραμμής bit (bit ή bit_b) σε υψηλή στάθμη και της άλλης σε χαμηλή
- Η χαμηλή τιμή οδηγεί στον κόρο το PMOS του αντιστροφέα και έτσι γίνεται η εγγραφή της τιμής στο κελί
- Έστω $A = 0$, $A_b = 1$, bit = 1, bit_b = 0



Νικόλαος Καββαδίας nkann@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων

Νικόλαος Καββαδίας nkann@uop.gr Σχεδίαση Ψηφιακών Κυκλωμάτων