

## Διευκρίνιση

**Τα περισσότερα από τα παρακάτω κείμενα είναι μεταφρασμένα κείμενα από το διαδίκτυο.**

**Οι εικόνες είναι από το διαδίκτυο.**

Ο μικροεπεξεργαστής (μP = microProcessor) είναι ένα λογικό κύκλωμα σε ένα τσιπ πυριτίου το οποίο εκτελεί λογικές και αριθμητικές πράξεις. Επίσης μετακινεί και επεξεργάζεται Bytes δεδομένων.

Ο μικροελεγκτής (μC = microController) είναι ένας μικροεπεξεργαστής με πρόσθετα κυκλώματα. Πχ.

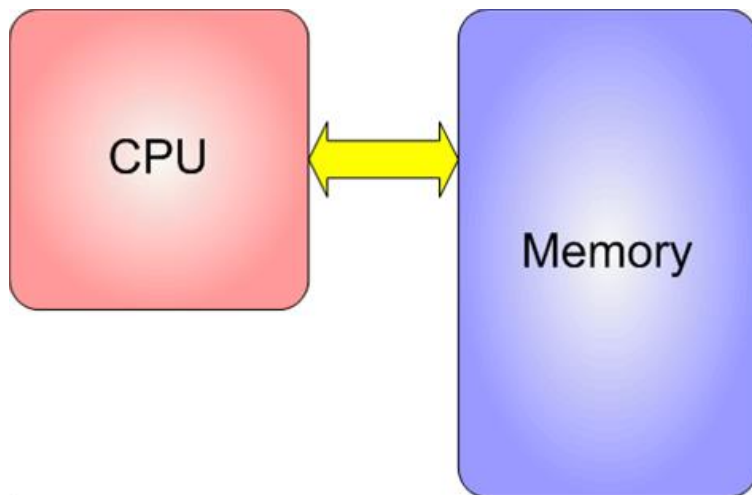
- Αναλογικούς σε Ψηφιακούς Μετατροπείς
- Ψηφιακούς σε Αναλογικούς Μετατροπείς
- Χρονιστές
- Κυκλώματα Σειριακής Επικοινωνίας
- Γεννήτριες PWM

**Ενσωματωμένο σύστημα:** είναι ένα σύστημα του οποίου η κύρια λειτουργία δεν είναι η υπολογιστική, αλλά αυτό ελέγχεται από έναν υπολογιστή που είναι ενσωματωμένο σ' αυτό. Το ενσωματωμένο σύστημα χρησιμοποιεί μικροελεγκτές. Σε αντίθεση με ηλεκτρονικούς υπολογιστές γενικής χρήσης, ένα ενσωματωμένο σύστημα έχει συνήθως ένα καλά καθορισμένο έργο που πρέπει να εκτελέσει αξιόπιστα και αποτελεσματικά - και με ελάχιστο κόστος, Οι υλοποιήσεις με μικροελεγκτή τείνουν να είναι ιδιαίτερα λιτές. Δεν υπάρχουν οι πολυτέλειες της προσωρινής αποθήκευσης πολλαπλών caching και συστήματα εικονικής μνήμης βασιζόμενα σε δίσκους. Υπάρχουν μόνο τα χαρακτηριστικά εκείνα που είναι σημαντικά για τη συγκεκριμένη εργασία.

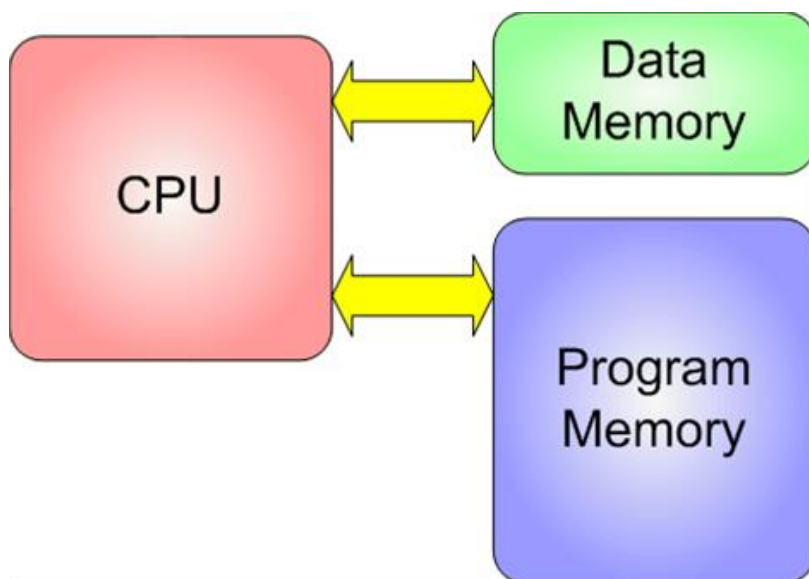
**Μια εντελώς διαφορετική κλίμακα.** Η μεγαλύτερη διαφορά μεταξύ των μικροελεγκτών και του υπολογιστή γενικού σκοπού είναι η καθαρή ποσότητα της διαθέσιμης μνήμης. Ο PIC 16F877 έχει μόνο 8K Bytes μνήμης Flash για την αποθήκευση του προγράμματος και 368 Bytes SRAM για την αποθήκευση δεδομένων Αυτό είναι πάνω από 100.000 φορές λιγότερο φυσικής μνήμης από ένα low-end PC!!!

## ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΜΙΚΡΟΪΠΟΛΟΓΙΣΤΩΝ

Η **Von Neumann (aka Princeton) αρχιτεκτονική** που αναπτύχθηκε για τον ENIAC χρησιμοποιεί την ίδια μνήμη και διαδρόμους τόσο για την αποθήκευση του προγράμματος όσο και για την αποθήκευση δεδομένων.



Η **Harvard αρχιτεκτονική** που χαρακτηρίστηκε από τον Harvard Mark 1 χρησιμοποιεί φυσικώς ξεχωριστή μνήμη και διαδρόμους για την αποθήκευση των δεδομένων και του προγράμματος.

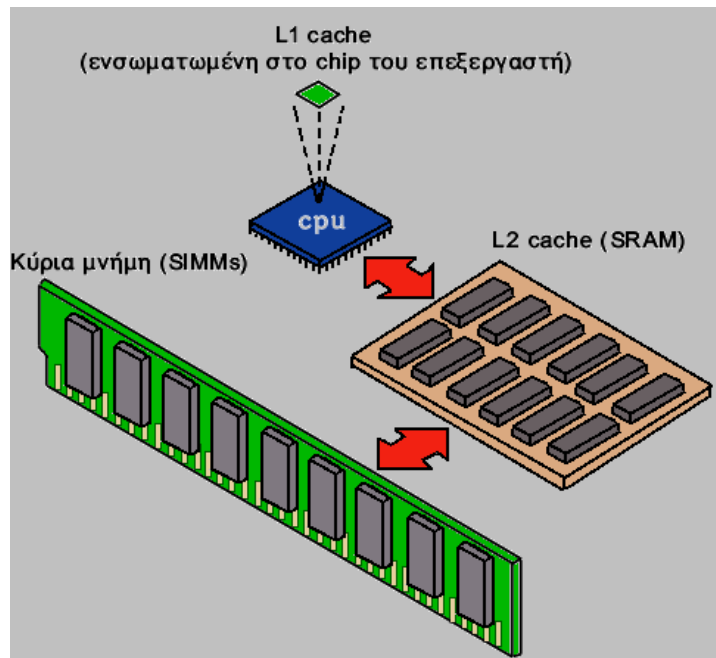


**Ποια είναι η καλύτερη;** Κάθε αρχιτεκτονική έχει τα πλεονεκτήματά της: εάν όλα τα άλλα είναι ίσα, η **Harvard αρχιτεκτονική** έχει το πλεονέκτημα στην απόδοση ενώ η **Von Neumann αρχιτεκτονική** είναι πιο ευέλικτη.

**Σύγχρονες Υβριδικές μορφές.** Αυτές τις μέρες, οι περισσότεροι υπολογιστές γενικής χρήσης (P.C. Mac ) είναι υβριδικές μορφές που δίνουν τα καλύτερα των δύο αρχιτεκτονικών. Μέσα στη CPU η λειτουργία βασίζεται στην **Harvard αρχιτεκτονική** χρησιμοποιώντας ξεχωριστές cache μνήμες για τις εντολές και τα δεδομένα για την μεγιστοποίηση της απόδοσης. Αλλά οι εντολές και τα δεδομένα φορτώνονται αυτόματα από ένα κοινό χώρο μνήμης (R.A.M.). Από τη σκοπιά του προγραμματισμού, οι υπολογιστές αυτοί φαίνεται να είναι καθαρά μηχανές **Von Neumann αρχιτεκτονικής**.

**Cache Μνήμη.** Προσοχή!!! Η cache μνήμη που αναφέρθηκε πιο πάνω είναι ένα σύνολο από καταχωρητές και έχουν την έννοια του cache (κρυφός) γιατί συνήθως δεν είναι προσβάσιμες από τον προγραμματιστή και βρίσκονται μέσα στον μP και μC. Δεν έχουν καμία σχέση με την cache μνήμη των μPs των Intel και AMD, η οποία αυτή μνήμη βρίσκεται επάνω στο πλακίδιο που φέρει και τον μP, αλλά εκτός αυτού. Η L1 Cache μνήμη βρίσκεται μέσα στο ίδιο chip αλλά είναι ανεξάρτητη από το C.P.U. μέρος. Βλέπε τις παρακάτω εικόνες:



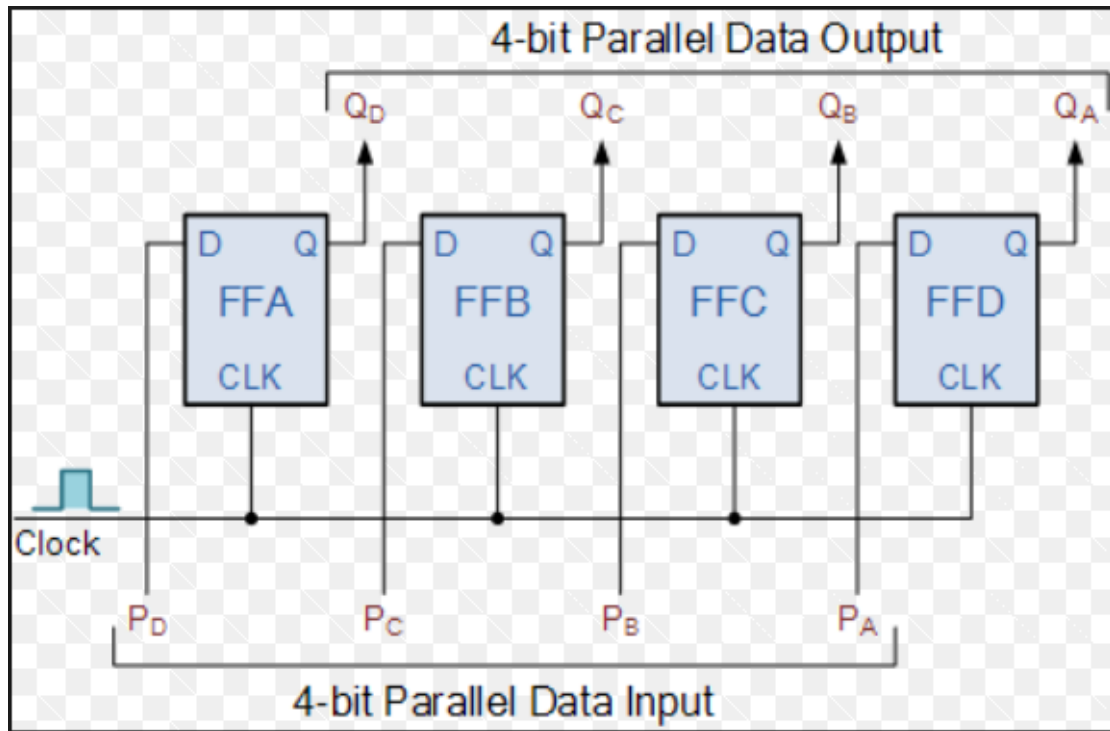


Σχετικές πληροφορίες για αυτού του είδους την cache μνήμη μπορείτε να βρείτε στον παρακάτω σύνδεσμο:

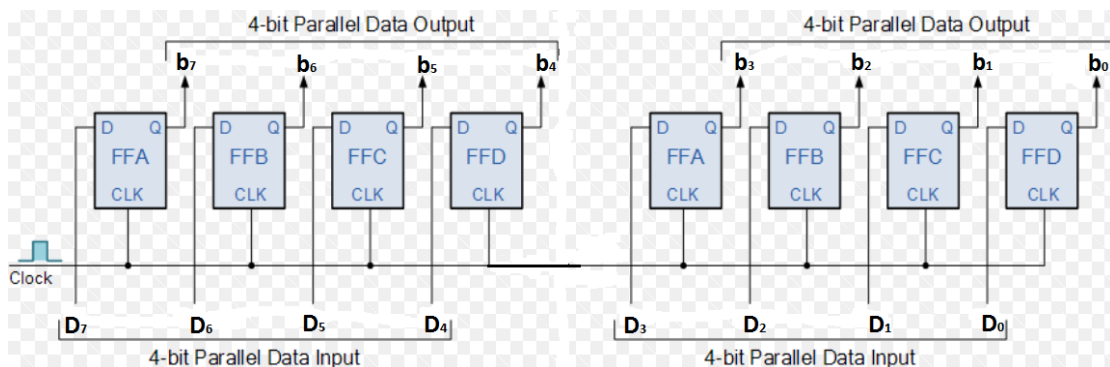
[http://el.wikipedia.org/wiki/%CE%9A%CF%81%CF%85%CF%86%CE%AE\\_%CE%BC%CE%BD%CE%AE%CE%BC%CE%B7](http://el.wikipedia.org/wiki/%CE%9A%CF%81%CF%85%CF%86%CE%AE_%CE%BC%CE%BD%CE%AE%CE%BC%CE%B7)

# ΚΑΤΑΧΩΡΗΤΕΣ

Στους καταχωρητές αποθηκεύουμε προσωρινά (συνήθως) κάποια δεδομένα. Η φόρτωση αυτών των δεδομένων γίνεται συνήθως παράλληλα και σπανιότερα σειριακά. Εκτός του chip των μικροεπεξεργαστών/μικροελεγκτών οι καταχωρητές μπορούν να υλοποιηθούν με την βοήθεια διαφόρων τύπων Flip-Flops. Στην παρακάτω εικόνα φαίνονται τέσσερα D-type Flip-Flops που υλοποιούν ένα καταχωρητή 4bits (= 1nibble). Και τα τέσσερα Flip-Flops έχουν κοινό clock.



Για να δημιουργήσουμε ένα καταχωρητή των 8bits (= 1Byte) μπορούμε να συνδέσουμε δύο από τους παραπάνω καταχωρητές με τον τρόπο που φαίνεται στο παρακάτω σχήμα:



Εάν χρησιμοποιούσαμε το Ο.Κ. 7474 που περιέχει δυο D-type Flip-Flops θα χρειαζόμασταν τέσσερα τέτοια Ολοκληρωμένα Κυκλώματα.

## Καταχωρητής

Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια

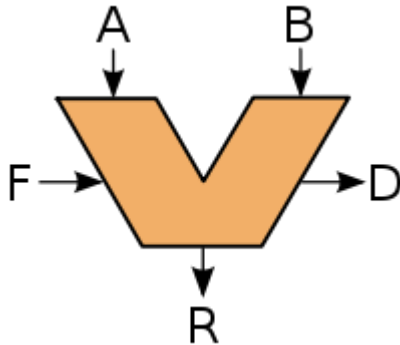
Στη [μηχανική υπολογιστών](#), ο **καταχωρητής** είναι τύπος μικρής αλλά πολύ γρήγορης μνήμης που βρίσκεται μέσα στο τσιπ του [επεξεργαστή](#). Η μνήμη αυτή χρησιμοποιείται για την βελτίωση της ταχύτητας εκτέλεσης των διαφόρων [προγραμμάτων](#), αφού σε αυτήν συνήθως αποθηκεύονται [δεδομένα](#) που χρησιμοποιούνται συνέχεια από τα προγράμματα. Στην περίπτωση αυτή ο καταχωρητής παρέχει πολύ γρήγορη πρόσβαση σε αυτά τα δεδομένα και έτσι το πρόγραμμα εκτελείται πιο γρήγορα. Οι περισσότεροι από τους σύγχρονους [ηλεκτρονικούς υπολογιστές](#) λειτουργούν σύμφωνα με την εξής λογική: μεταφέρουν δεδομένα από την [κεντρική μνήμη](#) στους καταχωρητές, κάνουν τις διάφορες πράξεις πάνω στα δεδομένα και στην συνέχεια μεταφέρουν το αποτέλεσμα από τους καταχωρητές πίσω στην κύρια μνήμη. Η τεχνική αυτή ονομάζεται load-store architecture.

Η ΚΜΕ περιέχει πολλούς καταχωρητές, από τους οποίους άλλοι είναι γενικής χρήσης, ενώ άλλοι επιτελούν μια συγκεκριμένη λειτουργία (καταχωρητές ειδικής χρήσης). Οι σημαντικότεροι από τους καταχωρητές ειδικής χρήσης είναι ο μετρητής προγράμματος (Program Counter, P.C.) και ο καταχωρητής εντολών (Instruction Register, I.R.). Ο μετρητής προγράμματος δείχνει την επόμενη εντολή που πρόκειται να εκτελεστεί, ενώ ο καταχωρητής εντολών περιέχει την εντολή που εκτελείται εκείνη τη στιγμή.

# ΑΡΙΘΜΗΤΙΚΗ και ΛΟΓΙΚΗ ΜΟΝΑΔΑ

## Αριθμητική και Λογική Μονάδα

Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια



Σχηματικός συμβολισμός της αριθμητικής/λογικής μονάδας.

Στη [μηχανική υπολογιστών](#) η **αριθμητική/λογική μονάδα (ALU)** είναι ένα [ψηφιακό κύκλωμα](#) το οποίο εκτελεί αριθμητικούς και λογικούς υπολογισμούς. Η ALU είναι θεμελιώδες δομικό στοιχείο της [Κ.Μ.Ε.](#) του υπολογιστή. Ακόμα και οι πιο απλοί [μικροεπεξεργαστές](#) έχουν μια για σκοπούς όπως η διαχείριση δεδομένων. Οι σύγχρονοι επεξεργαστές και οι μονάδες επεξεργασίας γραφικών, φιλοξενούν ισχυρές και πολύ πολύπλοκες ALU.

Ο μαθηματικός **Τζον Φον Νόιμαν** πρότεινε τον όρο ALU το 1945, όταν έγραψε μια αναφορά για την θεμέλια κατασκευή ενός καινούργιου τότε υπολογιστή, του [EDVAC](#). Η έρευνα γύρω από τις ALU εξακολουθεί να αποτελεί σημαντικό μέρος της επιστήμης των υπολογιστών.

Για κάθε ένα σύστημα, οι ALU είχαν διαφορετικούς σχεδιασμούς συχνά αρκετά πολύπλοκους. Έτσι καθιερώθηκε το αριθμητικό σύστημα του συμπληρώματος ως προς 2, μια αριθμητική παράσταση που καθιστά ευκολότερους τους υπολογισμούς για την ALU. Με αυτό το αριθμητικό σύστημα, η αφαίρεση επιτυγχάνεται με την πρόσθεση του αρνητικού αριθμού (π.χ. αντί για  $5-3$ ,  $5+(-3)$ ), και έτσι εξουδετερώνεται η ανάγκη για εξειδικευμένα κυκλώματα για την αφαίρεση.

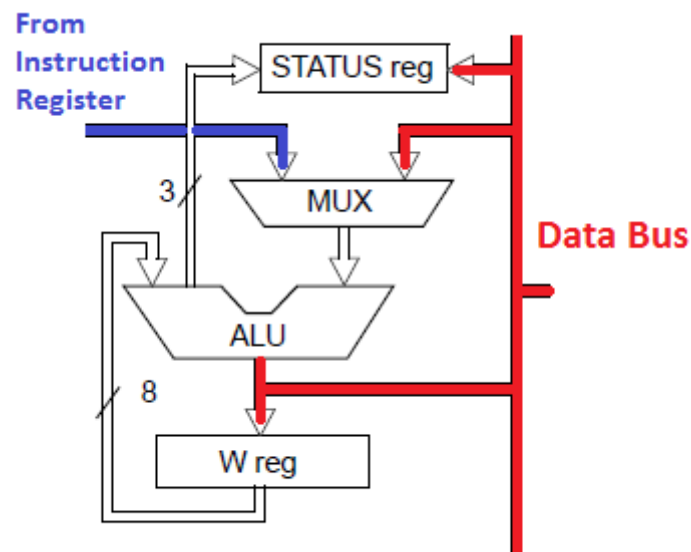
Οι περισσότερες ALU, μπορούν να εκτελέσουν τις παρακάτω πράξεις:

- Ακέραιες αριθμητικές πράξεις (πρόσθεση, αφαίρεση και μερικές φορές πολ/σμο και διαίρεση, αν και η υλοποίησή τους είναι ακριβή).
- Λογικές πράξεις (και, ή, όχι)

- Μετατόπιση ή περιστροφή μιας λέξης τόσο όσο ενός καθορισμένου αριθμού bits, προς τα αριστερά ή δεξιά, με πρόσημο ή χωρίς. Οι μετατοπίσεις μπορούν να υλοποιηθούν σαν πολλαπλασιασμοί με το 2 και ως διαιρέσεις με 2.

### Η Αριθμητική και Λογική Μονάδα του PIC16F877

Στο παρακάτω σχήμα φαίνεται ο τρόπος επικοινωνίας της ALU του PIC με το καταχωρητή Work και πώς αυτός ανατροφοδοτεί την ALU. Η ALU δέχεται δεδομένα από τον Καταχωρητή Εντολών (Instruction Register) και τον Διάδρομο Δεδομένων μέσω του πολυπλέκτη MUX. Η ALU εισάγει δεδομένα στον Διάδρομο Δεδομένων. Επίσης η ALU ενημερώνει τρεις σημαίες του Καταχωρητή Κατάστασης (STATUS register).



Οι Αριθμητικές πράξεις που εκτελούνται άμεσα με αυτή την ALU είναι οι

Πρόσθεση, και

Αφαίρεση



Οι Λογικές πράξεις που εκτελούνται άμεσα με αυτή την ALU είναι οι

Όχι (NOT),

Και (AND),

Ή (μη αποκλειστικό) (OR),

Αποκλειστικό Ή (XOR)

Η ALU αυτή εκτελεί επίσης:

Μεταφορές δεδομένων μεταξύ των καταχωρητών.

Αλλαγή της τιμής ενός συγκεκριμένου bit ενός δεδομένου.

Ολίσθηση των bits των δεδομένων προς τα αριστερά ή προς τα δεξιά.

Ανταλλαγή των nibbles του Byte ενός 8bits δεδομένου.

### **Η ALU ως ανεξάρτητο Ολοκληρωμένο Κύκλωμα**

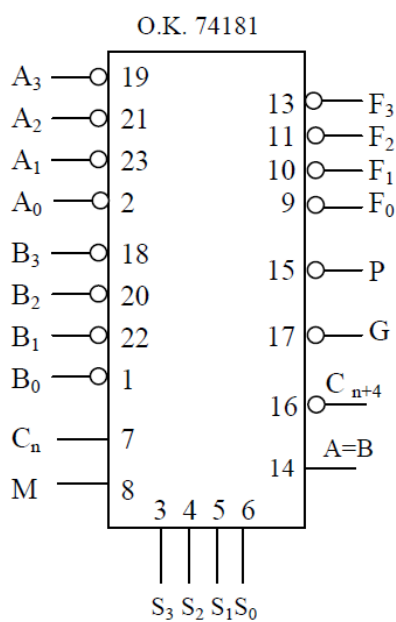
Στο παρακάτω σχήμα φαίνεται το λογικό σύμβολο της 4-bits ALU 74181 και ο τρόπος λειτουργία της. Από τις εισόδους  $M$  και  $S_3...S_0$  επιλέγεται κάθε φορά η πράξη που θα εκτελέσει το κύκλωμα, σύμφωνα με το πίνακα λειτουργίας του σχήματος. Τα  $A$  και  $B$  του πίνακα αναφέρονται στις 4-bits λέξεις (τελεστέους)  $A(A_3..A_0)$  και  $B(B_3..B_0)$ , ενώ το  $F(F_3..F_0)$  αποτελεί την έξοδο της ALU. Τα σύμβολα  $.$  και  $+$  αναφέρονται τέλος στις λογικές πράξεις AND και OR, αντίστοιχα.

Όταν  $M=1$ , επιλέγονται οι λογικές πράξεις που εκτελεί η ALU. Οι τιμές των εισόδων  $S_3...S_0$  καθορίζουν τότε μία από τις 16 συγκεκριμένες λογικές πράξεις που εκτελούνται, σύμφωνα με το πίνακα. Κάθε έξοδος της ALU σ' αυτή τη περίπτωση είναι συνάρτηση των αντίστοιχων δεδομένων που βρίσκονται στις εισόδους  $A$  και  $B$  και η είσοδος  $C_n$  (κρατούμενο εισόδου) αγνοείται.

Όταν  $M=0$ , επιλέγονται οι αριθμητικές πράξεις. Έτσι τυχόν προηγούμενο κρατούμενο στην είσοδο  $C_n$  λαμβάνεται πλέον υπ' όψη. Για πράξεις τέλος με τελεστέους μεγαλύτερους των 4 bits έχουμε τη δυνατότητα σύνδεσης περισσότερων της μιας ALU 74181 στη σειρά (αλυσιδωτά), με το κρατούμενο εξόδου ( $C_{n+4}$ ) της πρώτης (λιγότερο σημαντικής βαθμίδας) να συνδέεται απ' ευθείας με το κρατούμενο εισόδου ( $C_n$ ) της επόμενης (περισσότερο σημαντικής βαθμίδας).

Στη περίπτωση αυτή, οι είσοδοι επιλογής (M, S<sub>3</sub>..S<sub>0</sub>) όλων των ALU που χρησιμοποιούνται, πρέπει να έχουν τις ίδιες τιμές.

Οι είσοδοι A<sub>3</sub>..A<sub>0</sub> και B<sub>3</sub>..B<sub>0</sub> της ALU 74181 του σχήματος καθώς και η έξοδος F<sub>3</sub>..F<sub>0</sub> είναι ενεργές σε Low (active low operands). Η 74181 όμως μπορεί να χρησιμοποιηθεί και με ενεργές στο High εισόδους (active high operands) και έξοδο. Στη περίπτωση αυτή ο πίνακας του σχήματος τροποποιείται. Τις λεπτομέρειες αυτές για την 74181 θα τις βρείτε στα φύλλα δεδομένων του κατασκευαστή. Και σ' αυτή τη περίπτωση πάντως η ALU για M=0 εκτελεί λογικές πράξεις, ενώ για M=1, αριθμητικές.



α. Σχηματικό διάγραμμα

| Επιλογές<br>εισόδου<br>S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub> | M = 1<br>Λογικές<br>συναρτήσεις | M = 0<br>Αριθμητικές<br>πράξεις          |
|--|---------------------------------|--|
| 0 0 0 0  | F = A'                          | F=A μείον 1 συν C <sub>n</sub>           |
| 0 0 0 1  | F = A' + B'                     | F=A.B μείον 1 συν C <sub>n</sub>         |
| 0 0 1 0  | F = A' + B                      | F = A.B' μείον 1 συν C <sub>n</sub>      |
| 0 0 1 1  | F=1111                          | F = 1111 μείον 1 συν C <sub>n</sub>      |
| 0 1 0 0  | F = A' . B'                     | F = A συν (A+B') συν C <sub>n</sub>      |
| 0 1 0 1  | F = B'                          | F = A μείον B μείον 1 συν C <sub>n</sub> |
| 0 1 1 0  | F = A' . B'                     | F = A + B' συν C <sub>n</sub>            |
| 0 1 1 1  | F = A + B'                      | F = A συν (A+B) συν C <sub>n</sub>       |
| 0 0 0 0  | F = A' . B                      | F = A συν B συν C <sub>n</sub>           |
| 0 0 0 1  | F = A' . B                      | F = A . B' συν (A+B) συν C <sub>n</sub>  |
| 0 0 1 0  | F=B                             | F = A+B συν C <sub>n</sub>               |
| 0 0 1 1  | F=A+B                           | F = AB Μείον 1                           |
| 0 1 0 0  | F=0000                          | F = A συν A συν C <sub>n</sub>           |
| 0 1 0 1  | F = A . B'                      | F = A.B συν A συν C <sub>n</sub>         |
| 0 1 1 0  | F=A . B                         | F = A.B' συν A συν C <sub>n</sub>        |
| 0 1 1 1  | F=A                             | F = A συν C <sub>n</sub>                 |

β. Λειτουργία της ALU

**Σχήμα**

Η ALU του O.K. 74181