

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΜΗΧ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΟΜΕΑΣ ΥΛΙΚΟΥ ΚΑΙ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
Εργαστήριο Θεωρίας Κυκλωμάτων, Ηλεκτρονικών & Λογικού Σχεδιασμού

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ
ΛΟΓΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ

Γ.Φ. ΑΛΕΞΙΟΥ - Ε.Ζ. ΨΑΡΑΚΗΣ

Πάτρα 2009

‘Ακαδ. Έτος : 2009 - 2010

Εαρινό Εξάμηνο : 2009-2010

Όνοματεπώνυμο : _____

Α.Μ. : _____

Τμήμα : _____

Ημέρα: _____ **Ώρα:** _____

Αναλυτική Βαθμολογία Πρακτικού Μέρους:

	Βαθμός	Αξιολογητής	Υπογραφή Αξιολογητή
<i>Άσκηση 1</i>			
<i>Άσκηση 3</i>			
<i>Άσκηση 5</i>			
<i>Άσκηση 6</i>			
<i>Άσκηση 7</i>			
<i>Μ.Ο.</i>			

ΚΑΝΟΝΙΣΜΟΣ ΕΡΓΑΣΤΗΡΙΟΥ ΛΟΓΙΚΗΣ ΣΧΕΔΙΑΣΗΣ

1. Σε κάθε εργαστηριακή άσκηση θα πρέπει να προσέρχεστε διαβασμένοι. Θα γίνονται ερωτήσεις κατά τη διάρκεια της άσκησης και οι απαντήσεις θα λαμβάνονται υπόψη στη βαθμολόγηση της.
2. Το φυλλάδιο θα παραδίδεται ακριβώς μία εβδομάδα μετά από την ημερομηνία εκτέλεσης της άσκησης. Σε περίπτωση καθυστέρησης της παράδοσης, η συγκεκριμένη άσκηση θα βαθμολογείται με 0.
3. Σε περίπτωση που κάποιος συμπληρώσει δύο (2) απουσίες χάνει το δικαίωμα προσέλευσης και καλείται να επαναλάβει το σύνολο των εργαστηριακών ασκήσεων την επόμενη χρονιά. Τα ίδια ισχύουν όταν κάποιος έχει μία (1) απουσία και δεν προσέλθει στο επαναληπτικό εργαστήριο.
4. Δικαίωμα εκτέλεσης επαναληπτικού εργαστηρίου έχουν μόνο όσοι έχουν το πολύ μία (1) απουσία. Όποιος δεν έχει απουσία και δεν πρόλαβε να ολοκληρώσει κάποια ερωτήματα εργαστηριακών ασκήσεων ή θέλει να επαναλάβει κάποια για να τα εκτελέσει καλύτερα, μπορεί να προσέλθει στο επαναληπτικό εργαστήριο και για κάθε ερώτημα θα μετρήσει η καλύτερή του προσπάθεια.
5. Ο μέσος όρος των βαθμών των ασκήσεων πρέπει να είναι μεγαλύτερος ή ίσος του 5 ώστε να κατοχυρώσετε ασκήσεις, διαφορετικά καλείστε να τις επαναλάβετε όλες την επόμενη χρονιά. Σε αυτήν την περίπτωση δεν έχετε δικαίωμα να λάβετε μέρος στην τελική εξέταση.
6. Ο βαθμός της εξέτασης πρέπει να είναι μεγαλύτερος ή ίσος του 5, διαφορετικά καλείστε να την επαναλάβετε στην επόμενη εξεταστική περίοδο.
7. Ο τελικός βαθμός δίνεται από τη σχέση: $B_{\text{τελικός}} = (M \cdot O_{\text{ασκήσεων}} + B_{\text{εξέτασης}}) / 2$.
8. Στο τέλος κάθε άσκησης θα πρέπει να επιστρέφονται όλα τα υλικά (π.χ. καλώδια, αντιστάσεις, κλπ.) που δόθηκαν κατά τη διάρκειά της.
9. Στους χώρους του Εργαστηρίου απαγορεύονται το κάπνισμα, τα αναψυκτικά, οι καφέδες, τα σάντουιτς κλπ.

ΠΕΡΙΕΧΟΜΕΝΑ

ΑΣΚΗΣΕΙΣ

- 1. Λογικές Πύλες**
- 2. BCD Κώδικες / Αθροιστές**
- 3. Αφαιρέτες**
- 4. PLA' s**
- 5. Αριθμητικές/ Λογικές Μονάδες**
- 6. Καταχωρητές**
- 7. Μετρητές**

ΠΑΡΑΡΤΗΜΑ

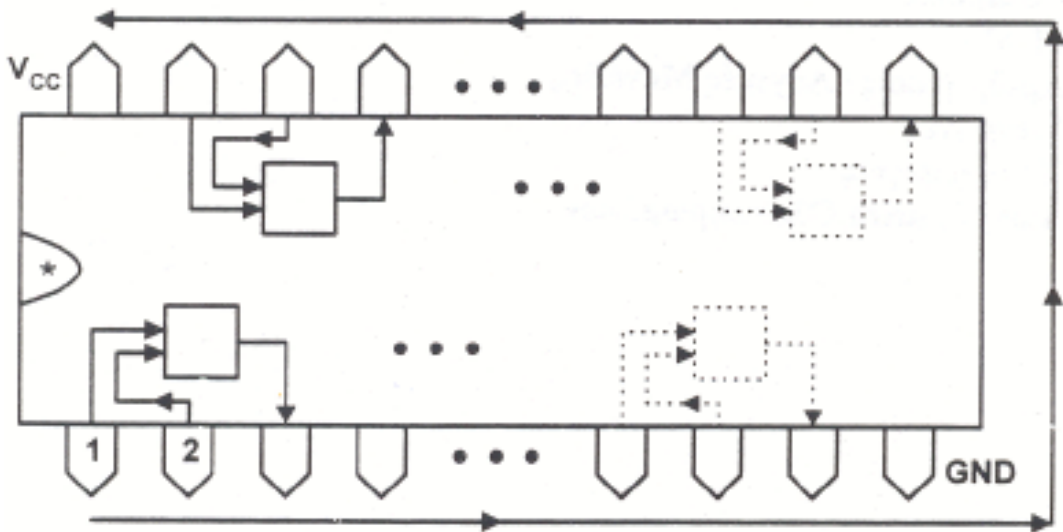
Διαγράμματα Ολοκληρωμένων

ΑΣΚΗΣΗ 1**ΗΜΕΡΟΜΗΝΙΑ:****ΘΕΜΑ: ΛΟΓΙΚΕΣ ΠΥΛΕΣ - ΣΥΝΑΡΤΗΣΕΙΣ BOOLE**

(Morris Mano, Κεφ. 1- 4).

ΘΕΩΡΙΑ:

Γενικά χαρακτηριστικά ολοκληρωμένων κυκλωμάτων.



Σχήμα 1

1. Όταν η εγκοπή (*) στο σχήμα 1 είναι αριστερά, η αρίθμηση για τους ακροδέκτες του chip γίνεται κατά τη φορά του βέλους (αντίθετα από τη φορά του ρολογιού).
2. Στα περισσότερα chips ο πάνω αριστερός ακροδέκτης είναι η τροφοδοσία (V_{CC}) και ο κάτω δεξιός είναι η γείωση (GND).
3. Οι είσοδοι - έξοδοι των πυλών έχουν την τοπολογία του σχήματος 1, εκτός από τη NOR (LS7402) που είναι ανάποδα (δείτε τα data sheets).

ΔΙΑΔΙΚΑΣΙΑ

1. Γράψτε και επαληθεύστε τους πίνακες αλήθειας των πυλών: NOT, AND, OR, NAND, NOR, XOR.

ΠΙΝΑΚΕΣ ΑΛΗΘΕΙΑΣ:

2. Απλοποιείστε και υλοποιείστε τις παρακάτω συναρτήσεις:

1^η Συνάρτηση

F (A, B, C, D) = Σ (1, 4, 5, 9, 10), που έχει τους ελαχιστόρους αδιάφορου τιμής:

d (A, B, C, D) = Σ (0, 2, 6, 8)

με τον ελάχιστο αριθμό πύλες NOR.

ΑΠΛΟΠΟΙΗΜΕΝΗ ΣΥΝΑΡΤΗΣΗ BOOLE:

2^η Συνάρτηση

$$F(A, B, C, D) = (A + B)(A' + D')(A' + B')$$

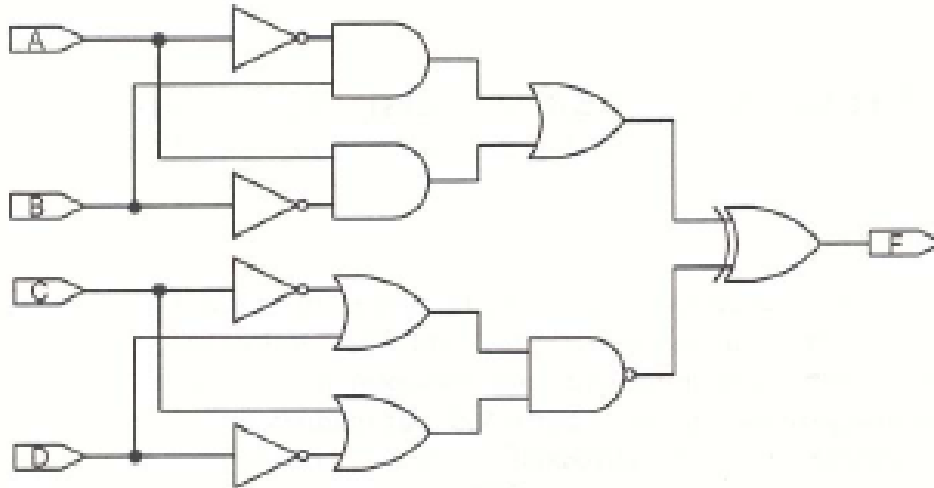
με όχι παραπάνω από τρεις πύλες NAND.

ΑΠΛΟΠΟΙΗΜΕΝΗ ΣΥΝΑΡΤΗΣΗ BOOLE:

3. Γιατί στις υλοποιήσεις συνδυαστικών κυκλωμάτων υπάρχει η συνήθεια να γίνεται η υλοποίηση με ένα μόνο είδος πύλης;

ΑΠΑΝΤΗΣΗ:

4. Αναλύστε και υλοποιήστε το κύκλωμα του σχήματος 2. Τι κάνει το κύκλωμα αυτό ;



Σχήμα 2

ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ

ΑΣΚΗΣΗ 2**ΗΜΕΡΟΜΗΝΙΑ:**

**ΘΕΜΑ: ΚΩΔΙΚΑΣ BCD . ΗΜΙΑΘΡΟΙΣΤΗΣ, ΑΘΡΟΙΣΤΗΣ -
HALF, FULL ADDERS (Morris Mano, Κεφ. 1- 4).**

ΘΕΩΡΙΑ**Κώδικας BCD.**

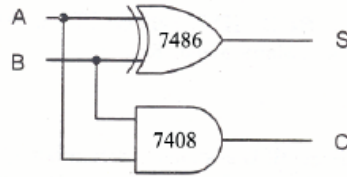
Ο κώδικας BCD είναι μία άμεση αντιστοιχία των δεκαδικών ψηφίων ενός αριθμού με το δυαδικό του ισοδύναμο. Με άλλα λόγια, για να παραστήσουμε έναν αριθμό σε BCD παίρνουμε τη δεκαδική του παράσταση και αντικαθιστούμε κάθε δεκαδικό του ψηφίο με το δυαδικό του ισοδύναμο. Έτσι ο δεκαδικός αριθμός 35 για παράδειγμα σε BCD είναι 0011 0101. Παρακάτω ακολουθεί πίνακας όλων των αριθμών 0 - 15 σε δεκαδική, δυαδική και BCD παράσταση:

ΔΕΚΑΔΙΚΗ	ΔΥΑΔΙΚΗ	BCD
00	0000	0000 0000
01	0001	0000 0001
02	0010	0000 0010
03	0011	0000 0011
04	0100	0000 0100
05	0101	0000 0101
06	0110	0000 0110
07	0111	0000 0111
08	1000	0000 1000
09	1001	0000 1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
14	1110	0001 0100
15	1111	0001 0101

Α.Μ.Α. Η Αλφαριθμητική Μονάδα Απεικόνισης (seven segment display) είναι έτοιμη πάνω στη συσκευή του εργαστηρίου και παίρνει σαν είσοδο την BCD παράσταση ενός αριθμού μεταξύ 0 και 9 και τον απεικονίζουν πάνω σ' ένα 7- segment display. Απαραίτητη η σύνδεση του "common" στη γη (GND).

Προσοχή!!! : Οι είσοδοι ABCD είναι ανάποδα (A = Λιγότερο Σημαντικό bit, D = Περισσότερο Σημαντικό bit)

Ημιαθροιστής (HALF ADDER, HA). Ο Ημιαθροιστής είναι ένα συνδυαστικό κύκλωμα δύο εισόδων A, B και δύο εξόδων S, C. Το κύκλωμα αυτό κάνει πρόσθεση των δύο εισόδων A, B (ενός bit) και δίνει σαν έξοδο το άθροισμά τους S (Sum) και το πιθανό κρατούμενο C (Carry). Το κύκλωμα και οι λογικές συναρτήσεις των δύο εξόδων του HA ακολουθούν.



Σχήμα 1

$$S = A' B + B' A = A \text{ XOR } B$$

$$C = AB$$

ΔΙΑΔΙΚΑΣΙΑ

1. Να σχεδιαστεί και να κατασκευαστεί συνδυαστικό κύκλωμα το οποίο να δίνει στην έξοδό του την BCD παράσταση μιας δυαδικής εισόδου που παίρνει τιμές 0-15 (4 bits). Για να βεβαιωθείτε ότι το κύκλωμά σας δουλεύει σωστά συνδέστε την έξοδο του κυκλώματός σας σε ένα LED και μία ΑΜΑ.

ΣΥΝΔΙΑΣΤΙΚΟ ΚΥΚΛΩΜΑ

2. Ένας **Αθροιστής 1 bit (1 bit FULL ADDER , FA)** είναι ένα συνδυαστικό κύκλωμα τριών εισόδων **A, B C_{in}** και δύο εξόδων **S, C_{out}** . Η έξοδος **S** είναι το άθροισμα των τριών εισόδων και το **C_{out}** είναι το πιθανό κρατούμενο της άθροισης. Η είσοδος **C_{in}** παίζει τον ρόλο του κρατούμενου που έρχεται από μία προηγούμενη βαθμίδα του **FA**.

α. Γράψτε τον πίνακα αλήθειας του FA και τις λογικές συναρτήσεις για τις δύο εξόδους.

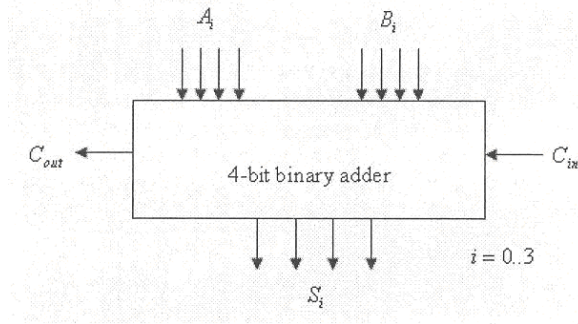
ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΚΑΙ ΛΟΓΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

β. Χρησιμοποιώντας δύο HA, σχεδιάστε και υλοποιείστε έναν FA.

ΚΥΚΛΩΜΑ FA

3. Χρησιμοποιείστε (1) chip 4008 (4 bit Full Adder) για να κατασκευάσετε ένα Αθροιστή (3) bit. Συνδέστε την έξοδό σας στο συνδυαστικό κύκλωμα της (ερώτησης 1) για να δείτε αν παίρνετε σωστά αποτελέσματα.

ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΥΚΛΩΜΑ ΑΘΡΟΙΣΤΗ ΤΩΝ 4 bits (4008)



ΚΥΚΛΩΜΑ ΑΘΡΟΙΣΤΗ 3 bits

ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ

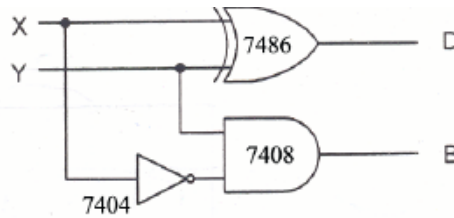
ΑΣΚΗΣΗ 3

ΗΜΕΡΟΜΗΝΙΑ:

**ΘΕΜΑ: ΗΜΙΑΦΑΙΡΕΤΗΣ, ΑΦΑΙΡΕΤΗΣ -
HALF, FULL SUBTRACTOR** (Morris Mano, Κεφ. 1- 4).

ΘΕΩΡΙΑ

ΗΜΙΑΦΑΙΡΕΤΗΣ (HALF SUBTRACTOR, HS). Ο HS είναι ένα συνδυαστικό κύκλωμα δύο εισόδων **X**, **Y** και δύο εξόδων **D** (Difference) και **B** (Borrow). Το κύκλωμα αυτό δίνει στην έξοδο του **D** τη διαφορά (**X**- **Y**) των δύο εισόδων και στο **B** μας λέει αν χρειαστήκαμε να δανειστούμε μια μονάδα για την αφαίρεση. Το κύκλωμα και οι λογικές συναρτήσεις των δύο εξόδων ακολουθούν.



Σχήμα 1

$$D = X' Y + X Y' = X \text{ XOR } Y$$

$$B = X' Y$$

ΔΙΑΔΙΚΑΣΙΑ

1. Υποθέστε ότι έχετε ένα προσημασμένο αριθμό 4- bits της μορφής $sX_2X_1X_0$. Όταν $s=1$ τότε ο αριθμός σας είναι αρνητικός και παρίσταται με "συμπλήρωμα ως προς 2" (**2' s complement**).

Σχεδιάστε και κατασκευάστε συνδυαστικό κύκλωμα MONO με απλές πύλες (όχι MSI κυκλώματα) που να έχει σαν είσοδο ένα προσημασμένο αριθμό 4- bits $sX_2X_1X_0$ και σαν εξόδους $sY_2Y_1Y_0$. Το κύκλωμα πρέπει να έχει τις ακόλουθες ιδιότητες:

α. Όταν $s=0$ τότε $Y_i = X_i$

β. Όταν $s=1$ (αρνητικός αριθμός), τότε ο συνδυασμός των Y_i παριστάνει το 2' s complement της εισόδου.

Με άλλα λόγια η έξοδος του κυκλώματός σας είναι η παράσταση της εισόδου σε "Πρόσημο και Μέτρο" (Sign-Magnitude) μορφή.

Συνδέστε τις εξόδους σας σε μία AMA και το πρόσημο s σε μία LED για να βεβαιωθείτε ότι το κύκλωμά σας δουλεύει σωστά.

ΣΥΝΔΙΑΣΤΙΚΟ ΚΥΚΛΩΜΑ

2. Ένας **Αφαιρέτης (FULL SUBTRACTOR, FS)** είναι ένα συνδυαστικό κύκλωμα τριών εισόδων **X, Y, B_{in}** και δύο εξόδων **D, B_{out}**. Η έξοδος **D** είναι η διαφορά των τριών εισόδων (**X - Y - B_{in}**) και το **B_{out}** είναι το πιθανό δανεισμένο bit.

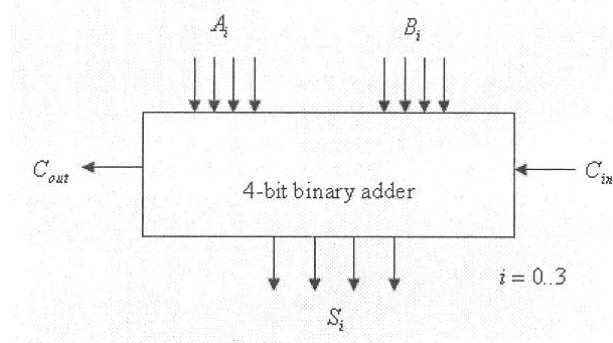
α. Γράψτε τον πίνακα αλήθειας του **FS** και τις λογικές συναρτήσεις για τις δύο εξόδους.

ΠΙΝΑΚΑΣ ΑΛΗΘΕΙΑΣ ΚΑΙ ΛΟΓΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

β. Χρησιμοποιώντας δύο **HS** σχεδιάστε και υλοποιήστε έναν **FS**.
ΚΥΚΛΩΜΑ FS

3. 3. Χρησιμοποιήστε (1) chip 4008 (4 bit Full Adder) για να κατασκευάσετε ένα Αφαιρέτη (4) bit. Συνδέστε την έξοδό σας στο συνδυαστικό κύκλωμα της (ερώτησης 1) για να δείτε αν παίρνετε σωστά αποτελέσματα.

ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΥΚΛΩΜΑ ΑΘΡΟΙΣΤΗ 4 bits (4008)



ΚΥΚΛΩΜΑ ΑΦΑΙΡΕΤΗ 4 bits

ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ.

ΑΣΚΗΣΗ 4

ΗΜΕΡΟΜΗΝΙΑ:

ΘΕΜΑ: ΠΡΟΓΡΑΜΜΑΤΙΖΟΜΕΝΗ ΛΟΓΙΚΗ -

PROGRAMMABLE LOGIC ARRAYS, PLA (Morris Mano, Γ' Έκδοση, Σελ. 382 - 387).

ΘΕΩΡΙΑ

PLA

Το **PLA** είναι ένα συνδυαστικό κύκλωμα που έχει τη δυνατότητα να προγραμματιστεί. Ο προγραμματισμός του επιτυγχάνεται με την καταστροφή ορισμένων συνδέσμων του αρχικού κυκλώματος. Τα **PLA** μπορούν να υλοποιήσουν συναρτήσεις Boole που έχουν εκφραστεί σε μορφή αθροίσματος γινομένων και συνήθως προτιμούνται από άλλες υλοποιήσεις όταν οι συναρτήσεις έχουν πολλούς αδιάφορους όρους. Το διάγραμμα ενός **PLA** n εισόδων και m εξόδων φαίνεται στο σχήμα 1.

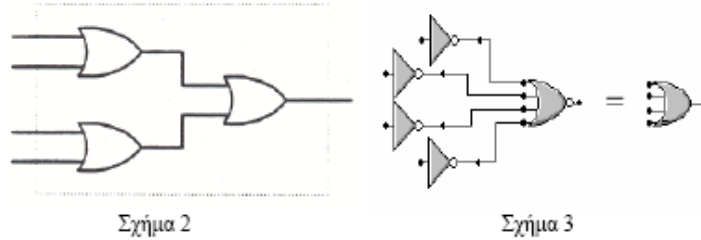


Σχήμα 1

Παρατηρείστε από το παραπάνω σχήμα ότι τα **PLA** αποτελούνται από έναν array πυλών AND όπου οδηγούμε όλες τις εισόδους. Οι έξοδοι των πυλών αυτών στη συνέχεια οδηγούνται σ' έναν Πίνακα Πυλών OR, οι έξοδοι των πυλών αυτών είναι οι συναρτήσεις που θέλουμε να υλοποιήσουμε. Πιο συγκεκριμένα τώρα, ας υποθέσουμε ότι έχουμε n εισόδους και θέλουμε να υλοποιήσουμε m συναρτήσεις Boole που είναι υπό τη μορφή αθροίσματος γινομένων. Ας υποθέσουμε επίσης ότι σε όλες αυτές τις συναρτήσεις τα διαφορετικά γινόμενα είναι k . Τότε θα χρειαστούμε k πύλες **AND** και m πύλες **OR**. Κάθε πύλη **AND** θα είναι $2 \cdot n$ εισόδων και αυτό γιατί σε κάθε πύλη οδηγούνται όλες οι εισοδοί και τα συμπληρώματά τους. Κάθε πύλη **OR** τώρα έχει k εισόδους. Όπως μπορούμε να δούμε και από το Σχ. 1 οι έξοδοι των **OR** περνούν και από πύλες **NOT** δίνοντάς μας την δυνατότητα να πάρουμε στην τελική έξοδο είτε την έξοδο της OR ή το συμπλήρωμά της. Το βασικό σε αυτή την συνδεσμολογία είναι ότι οι εισοδοί οδηγούνται στις αντίστοιχες πύλες μέσω διασυνδέσεων που έχουμε την δυνατότητα να κόψουμε (προγραμματισμός του **PLA**).

ΔΙΑΔΙΚΑΣΙΑ

1.. Σχεδιάστε και υλοποιείστε ένα PLA 2 εισόδων ($n = 2$), 4 εξόδων ($m = 4$) και 4 γινομένων ($k = 4$). Στη θέση των πυλών AND χρησιμοποιείστε πύλες NAND τεσσάρων εισόδων (7420) σε συνδυασμό με πύλες NOT (7404). Στη θέση των πυλών OR τεσσάρων εισόδων χρησιμοποιείστε συνδυασμό τριών πυλών OR δύο εισόδων (7432) σύμφωνα με τη συνδεσμολογία του Σχ.2.



ΚΥΚΛΩΜΑ PLA, $n=2, k=4, m=4$

2. Χρησιμοποιώντας το κύκλωμα του ερωτήματος 1 και αφαιρώντας τις κατάλληλες διασυνδέσεις κατασκευάστε συνδυαστικό κύκλωμα που να δίνει στην έξοδό του το τετράγωνο των αριθμών 0, 1, 2, 3. Γράψτε τις συναρτήσεις Boole των τεσσάρων εξόδων σε μορφή κατάλληλη για PLA και δώστε τον πίνακα προγραμματισμού του PLA.

ΣΥΝΑΡΤΗΣΕΙΣ BOOLE ΚΑΙ ΠΙΝΑΚΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

3. Χρησιμοποιώντας το κύκλωμα του ερωτήματος 1 και αφαιρώντας τις κατάλληλες διασυνδέσεις κατασκευάστε συνδυαστικό κύκλωμα που να δίνει στην έξοδό του την τετραγωνική ρίζα των αριθμών 0, 2, 8, 10. Να παραστήσετε τη ρίζα χρησιμοποιώντας 4 bits από τα οποία τα δύο πρώτα είναι για το ακέραιο μέρος και τα υπόλοιπα δύο για το κλασματικό. Δηλαδή αν η έξοδός σας είναι $F_3F_2.F_1F_0$ τότε αυτό συμβολίζει τον δεκαδικό αριθμό $2F_3 + F_2 + 2^{-1}F_1 + 2^{-2}F_0$. Σαν προσέγγιση 4 bits να θεωρηθεί εκείνος ο αριθμός που βρίσκεται πιο κοντά (μεγαλύτερος ή μικρότερος) στην πραγματική ρίζα. Π.χ. η τετραγωνική ρίζα του 12 είναι 3.4641.... και προσεγγίζεται με 4 bits από τον δυαδικό 11.10 (=3.5).

Γράψτε τις προσεγγίσεις των τεσσάρων αριθμών σε δεκαδική και δυαδική μορφή.

Γράψτε τις προσεγγίσεις των τεσσάρων αριθμών σε δεκαδική και δυαδική μορφή. Γράψτε τις συναρτήσεις Boole των τεσσάρων εξόδων σε μορφή κατάλληλη για PLA και δώστε τον πίνακα προγραμματισμού του PLA.

ΔΕΚΑΔΙΚΗ ΚΑΙ ΔΥΑΔΙΚΗ ΠΑΡΑΣΤΑΣΗ ΤΕΤΡΑΓΩΝΙΚΩΝ ΡΙΖΩΝ:

ΣΥΝΑΡΤΗΣΕΙΣ BOOLE ΚΑΙ ΠΙΝΑΚΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ:

ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ

ΑΣΚΗΣΗ 5**ΗΜΕΡΟΜΗΝΙΑ:**

**ΘΕΜΑ: ΑΡΙΘΜΗΤΙΚΗ_ΛΟΓΙΚΗ ΜΟΝΑΔΑ
(ARITHMETIC LOGIC UNIT, ALU)
ΜΕΤΑΡΤΡΟΠΕΑΣ ΨΗΦΙΚΟΥ ΣΕ ΑΝΑΛΟΓΙΚΟ
(DIGITAL TO ANALOG CONVERTER, DAC)**

ΘΕΩΡΙΑ**ALU**

Η ALU είναι ένα συνδυαστικό κύκλωμα δύο εισόδων 4 bits που μπορεί να εκτελέσει διάφορες αριθμητικές και λογικές πράξεις. Το είδος της πράξης καθορίζεται από τον συνδιασμό που θα δώσουμε στα ποδαράκια επιλογής (συμβουλευτείτε τα data sheets για το 74181).

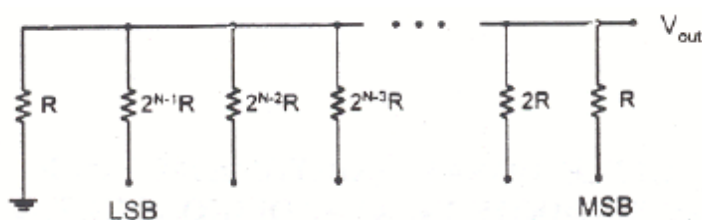
DAC

Ο DAC είναι ένα κύκλωμα που δέχεται σαν είσοδο μια ψηφιακή λέξη και βγάζει στην έξοδο του μιά αναλογική τάση ανάλογη του αριθμού που παριστάνει η ψηφιακή λέξη. Έτσι, αν έχουμε μια ψηφιακή λέξη N- bits $A_{N-1} \dots A_0$, ο DAC έχει σαν έξοδο μια τάση η τιμή της οποίας δίνεται από τον τύπο

$$V_{out} = (A_{N-1}2^{-1} + \dots + A_02^{-N}) V_0 \quad (1)$$

Όπως βλέπουμε, το περισσότερο σημαντικό ψηφίο A_{N-1} (MSB) είναι αυτό που αντιστοιχεί στην τάση $V_0/2$, το δε λιγότερο σημαντικό (LSB) αντιστοιχεί στην τάση $V_0/2^N$. Αν για παράδειγμα έχουμε λέξη 4 bits και $V_0 = 16$ τότε μπορούμε χρησιμοποιώντας την (1) να δούμε ότι το V_{out} έχει τιμή ίση με τον αριθμό που παριστάνει η δυαδική λέξη.

Υπάρχουν πολλά κυκλώματα που υλοποιούν την διαδικασία DAC. Υπάρχουν μάλιστα και ολοκληρωμένα που κάνουν αυτή τη δουλειά. Στο παρακάτω σχήμα δίνουμε μια απλή συνδεσμολογία (τύπου κλίματος) για μετατροπή DAC μιας λέξης N bits.



Σχήμα 1

Για άλλες συνδεσμολογίες και περισσότερες λεπτομέρειες ανατρέξτε στο βιβλίο Millman-Χαλκιάς Σελ. 309.

ΔΙΑΔΙΚΑΣΙΑ

1. Ελέγξτε τις αριθμητικές και λογικές λειτουργίες της ALU (74181).
2. Χρησιμοποιώντας την ALU σχεδιάστε και υλοποιήστε συνδυαστικό κύκλωμα το οποίο να συγκρίνει δύο αριθμούς τεσσάρων bits.

ΣΥΝΔΙΑΣΤΙΚΟ ΚΥΚΛΩΜΑ

3. Υλοποιείτε έναν DAC τύπου κλίματος για ψηφιακή λέξη 2 bits. Υποθέστε ότι η τάση στις δύο εισόδους είναι αντίστοιχα V_0 και V_1 .

a. Γράψτε τη σχέση που δίνει την τάση εξόδου V_{out} συναρτήσει των δύο τάσεων εισόδου V_0 και V_1 . Ποιο κατά τη γνώμη σας είναι το βασικότερο μειονέκτημα της συνδεσμολογίας αυτής;

ΣΧΕΣΗ ΕΙΣΟΔΩΝ-ΕΞΟΔΟΥ:

ΜΕΙΟΝΕΚΤΗΜΑ ΚΥΚΛΩΜΑΤΟΣ:

β. Με την βοήθεια του παλμογράφου μετρήστε τις εξόδους του κυκλώματός σας για όλους τους δυνατούς συνδυασμούς εισόδων.

Γράψτε τις τιμές που μετρήσατε.

ΜΕΤΡΗΣΕΙΣ:

4. Τροφοδοτείστε τις εισόδους A_0 και B_0 της **ALU** με δύο τετραγωνικούς παλμούς με λόγο συχνότητας $1/2$ ($f_1/f_0 = 1/2$). Επιλέξτε την λειτουργία "ΑΡΙΘΜΗΤΙΚΗ ΠΡΟΣΘΕΣΗ". Βεβαιωθείτε για τη σωστή λειτουργία της ALU ελέγχοντας τις εξόδους F_0 και F_1 . Σχεδιάστε τις κυματομορφές εξόδου.

ΚΥΜΑΤΟΜΟΡΦΕΣ ΕΞΟΔΟΥ

5. Συνδέστε τις δύο εξόδους του προηγούμενου ερωτήματος στις εισόδους του DAC. Με τον παλμογράφο παρατηρήστε την τάση εξόδου, σχεδιάστε την και εξηγήστε την.

ΚΥΜΑΤΟΜΟΡΦΗ ΕΞΟΔΟΥ

ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ.

ΑΣΚΗΣΗ 6

ΗΜΕΡΟΜΗΝΙΑ:

ΘΕΜΑ: ΚΑΤΑΧΩΡΗΤΕΣ - REGISTERS

(M. Mano, Γ' Έκδοση, Κεφ. 6, σελ. 301 - 315)

ΘΕΩΡΙΑ.

Ένας καταχωρητής (register) είναι μια ομάδα από δυαδικά κύτταρα αποθήκευσης (flip-flops) τα οποία όπως ξέρουμε είναι κατάλληλα για την αποθήκευση δυαδικών πληροφοριών. Ένας καταχωρητής των "n" bits περιέχει "n" flip-flops και επομένως μπορεί να αποθηκεύσει κάθε πληροφορία των "n" bits. Με την ευρύτερη έννοια, ένας καταχωρητής αποτελείται από ένα σύνολο flip-flops τα οποία όπως είπαμε κρατούν τις δυαδικές πληροφορίες και από ένα σύνολο πυλών για την επίτευξη της μεταφοράς των πληροφοριών.

Η μεταφορά νέων πληροφοριών μέσα σ' ένα καταχωρητή λέγεται "φόρτωση" του καταχωρητή. Ανάλογα με τον τρόπο μεταφοράς των δεδομένων στον καταχωρητή διακρίνουμε δύο τύπους καταχωρητών:

- (1) Τους καταχωρητές παράλληλης φόρτωσης (parallel load registers).
- (2) Τους καταχωρητές ολίσθησης (shift registers).

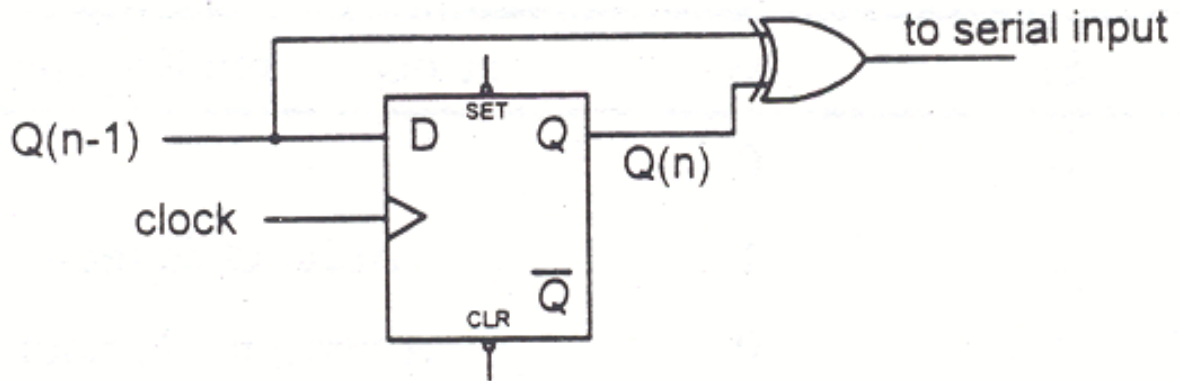
Στον τύπο (1) των καταχωρητών όλα τα bits του καταχωρητή φορτώνονται την ίδια χρονική στιγμή, δηλαδή η φόρτωση γίνεται παράλληλα. Στον τύπο (2) των καταχωρητών η πληροφορία διαδίδεται κατά μήκος του καταχωρητή. Ένας τέτοιος καταχωρητής αποτελείται από μία αλυσίδα από flip-flop συνδεδεμένα στη σειρά, με την έξοδο του ενός να τροφοδοτεί την είσοδο του γειτονικού του. Όλα τα flip-flops παίρνουν ένα κοινό ρολόι και το οποίο προκαλεί την ολίσθηση από την μία βαθμίδα στην επόμενη. Σ' ένα τέτοιο καταχωρητή η είσοδος της πρώτης βαθμίδας ονομάζεται "σειριακή είσοδος" (serial input) ενώ η έξοδος της τελευταίας βαθμίδας, ονομάζεται "σειριακή έξοδος" (serial output).

ΔΙΑΔΙΚΑΣΙΑ

(1): Κατασκευάστε έναν καταχωρητή ολίσθησης 4 bits. Βεβαιωθείτε για την σωστή λειτουργία του κυκλώματός σας. Σχεδιάστε το κύκλωμα σας στον παρακάτω χώρο.

(2) Συνδέστε την "σειριακή έξοδο" του παραπάνω κυκλώματος στην "σειριακή είσοδο" του. Κάντε PRESET την τελευταία βαθμίδα του κυκλώματος σας και επιτρέψτε την "δυναμική" λειτουργία του. Εξηγήστε την λειτουργία του κυκλώματος. Που έχουν εφαρμογή τέτοιου είδους κυκλώματα;

(3): Τροποποιήστε την τελευταία βαθμίδα του κυκλώματος όπως φαίνεται στο Σχήμα 1.



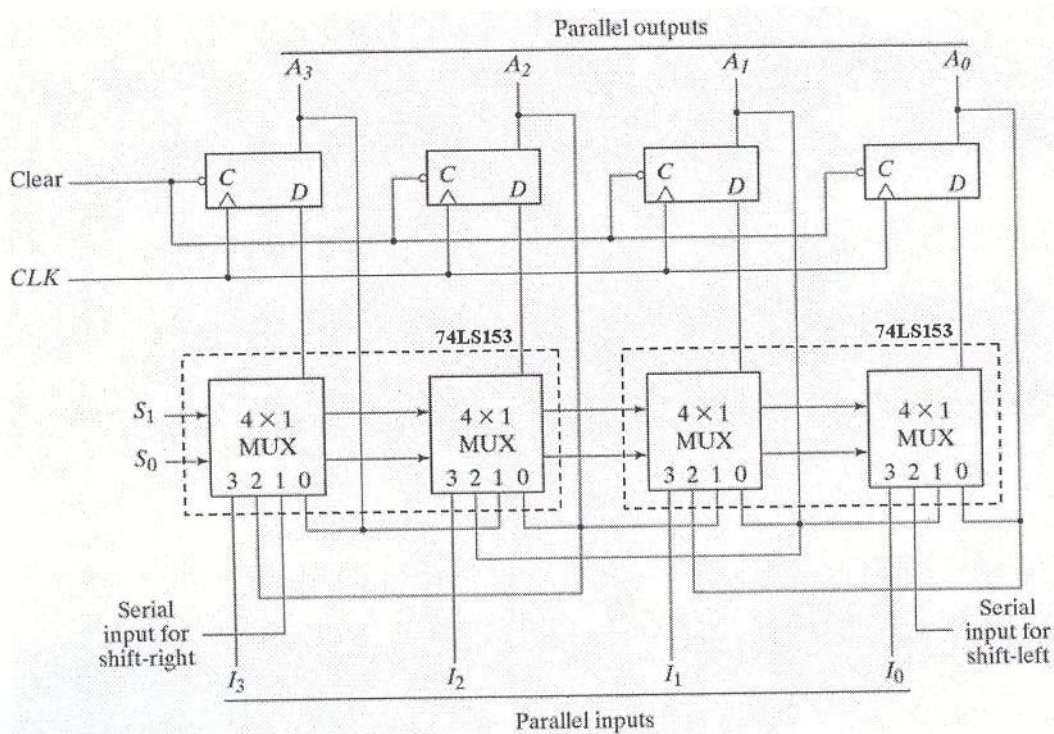
Σχήμα 1

Καταγράψτε τις εξόδους $Q(1) - Q(4)$ στον παρακάτω χώρο και εξηγήστε την λειτουργία του κυκλώματος.

(4): Να κατασκευάσετε κύκλωμα καταχωρητή που να εκτελεί τις λειτουργίες που φαίνονται στον πίνακα:

S_1	S_2	Λειτουργία του Καταχωρητή
0	0	-
0	1	Δεξιά Ολίσθηση
1	0	Αριστερή Ολίσθηση
1	1	Παράλληλη Φόρτωση

Σχηματικό της υλοποίησης



(5): Χρησιμοποιώντας το ολοκληρωμένο 74194 (4-bit bidirectional shift register), κατασκευάστε ένα κύκλωμα που θα εκτελεί σειριακή πρόσθεση. Το αποτέλεσμα της πράξεως να αποθηκεύεται στον καταχωρητή A. Σχεδιάστε το κύκλωμα.:

ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ

ΑΣΚΗΣΗ 7

ΗΜΕΡΟΜΗΝΙΑ:

ΘΕΜΑ: ΚΥΚΛΩΜΑΤΑ ΜΕΤΡΗΤΩΝ - COUNTERS

(M. Mano, Γ' Έκδοση, Κεφ. 6, σελ. 316 – 337)

ΑΠΑΡΑΙΤΗΤΑ ΥΛΙΚΑ.

Δύο (2) chip 7476 (JK Flip- Flop).

Δύο (2) chip 7432 (OR πύλες).

Δύο (2) chip 7400 (NAND πύλες).

Δύο (2) chip 7408 (AND πύλες).

Δύο (2) chip 7404 (NOT πύλες).

ΜΕΡΟΣ Α': ΘΕΩΡΙΑ

Οι ΜΕΤΡΗΤΕΣ είναι ακολουθιακά κυκλώματα τα οποία συνήθως χρησιμοποιούνται σαν βασικοί λίθοι στην σχεδίαση και υλοποίηση πολυπλοκότερων λογικών κυκλωμάτων. Βασικές λειτουργίες στις οποίες χρησιμοποιούνται είναι η μέτρηση και διαίρεση συχνότητας, η μέτρηση του χρόνου, οι αριθμητικές πράξεις, κ.λ.π.

Τα κυκλώματα των ΜΕΤΡΗΤΩΝ υλοποιούνται με χρήση JK, RS, T ή D Flip- Flops και μπορούν να διαιρεθούν σε δύο βασικές κατηγορίες.

Στους ασύγχρονους (λέγονται και σειριακοί ή ripple ΜΕΤΡΗΤΕΣ).

Στους σύγχρονους (λέγονται και παράλληλοι ΜΕΤΡΗΤΕΣ).

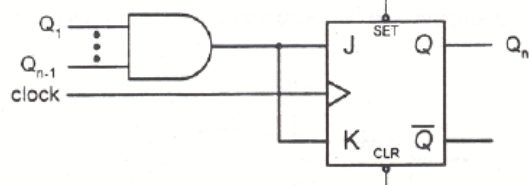
Στους σύγχρονους ΜΕΤΡΗΤΕΣ, οι έξοδοι όλων των flip-flops αλλάζουν κατάσταση ταυτόχρονα, σε αντίθεση με τους ασύγχρονους όπου η είσοδος κάθε FF σκανδαλίζεται από την αλλαγή κατάστασης της εξόδου του προηγούμενου FF.

Ανεξάρτητα από την κατηγορία που ανήκει, ένας ΜΕΤΡΗΤΗΣ είναι ένα ακολουθιακό κύκλωμα το οποίο επαναλαμβάνει μία ακολουθία N προκαθορισμένων, από τον σχεδιαστή, καταστάσεων όπου N το modulo του ΜΕΤΡΗΤΗ.

ΜΕΡΟΣ Α΄ : ΔΙΑΔΙΚΑΣΙΑ

(A.1) Με την βοήθεια του βασικού ακολουθιακού κυκλώματος (στοιχείου) του σχήματος 2, να σχεδιαστεί και να υλοποιηθεί **σύγχρονος δυαδικός μετρητής 4-bit**.

Εξηγήστε την λειτουργία του βασικού κυκλώματος του Σχ. 2.



Σχήμα 2

Σχεδιάστε εδώ το κύκλωμα του μετρητή:

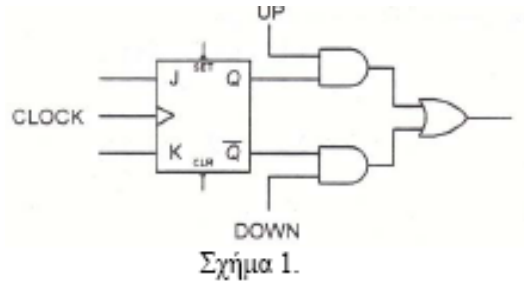
(A.2) Για συχνότητα ωρολογιακού παλμού (clock) 1 KHz μετρείστε με την χρήση παλμογράφου, τη συχνότητα της εξόδου Q_3 . Σχεδιάστε τις κυματομορφές:

(α): clock:

(b): Q_3 :

(A.3) Με την βοήθεια του βασικού ακολουθιακού κυκλώματος του σχήματος 1 να σχεδιαστεί και να υλοποιηθεί **ασύγχρονος up/down δυαδικός μετρητής 4-bit**.

Εξηγήστε την λειτουργία του βασικού κυκλώματος του σχήματος 1.



Σχεδιάστε εδώ το κύκλωμα του μετρητή.

(A.4) Για συχνότητα ωρολογιακού παλμού (clock) 1 KHz μετρήστε με χρήση παλμογράφου, την συχνότητα της εξόδου Q4. Να σχεδιαστούν οι κυματομορφές.

(α): clock

(β): Q4

ΜΕΡΟΣ Β : ΘΕΩΡΙΑ**Modulo N Μετρητής**

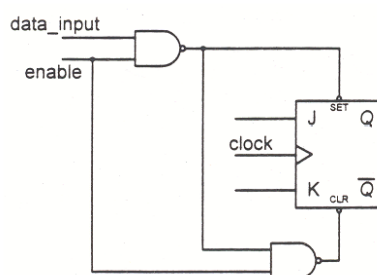
Ένας modulo N μετρητής είναι ένας μετρητής (σύγχρονος ή ασύγχρονος), ο οποίος έχει N διαφορετικές καταστάσεις. Ο παραπάνω ορισμός του modulo N μετρητή, είναι πολύ γενικός χωρίς κανένα περιορισμό στο N. Στην πράξη ένας modulo N μετρητής συνήθως κατασκευάζεται συνδέοντας σειριακά m ΜΕΤΡΗΤΕΣ modulo $N_i, i=1, \dots, m$ με τρόπο ώστε:

$$N = N_1 * N_2 * N_3 * \dots * N_m.$$

Π.χ. Ένας modulo 105 μετρητής μπορεί να υλοποιηθεί με σειριακή σύνδεση τριών μετρητών modulo 3, 5 και 7 αντίστοιχα, μια και $3 * 5 * 7 = 105$.

Προγραμματιζόμενοι μετρητές (programmable counters).

Προγραμματιζόμενος counter, είναι κάθε counter του οποίου το modulo μπορεί να τροποποιηθεί με την βοήθεια κάποιων γραμμών ελέγχου και κάποιων γραμμών δεδομένων.

**ΜΕΡΟΣ Β : ΔΙΑΔΙΚΑΣΙΑ**

(B.1) Με χρήση του βασικού ακολουθιακού κυκλώματος του σχήματος 3 να σχεδιαστεί και να υλοποιηθεί προγραμματιζόμενος counter 4-bit, ο οποίος θα μπορεί να τοποθετείται σε οποιαδήποτε επιθυμητή τιμή μεταξύ 0 και 15 με την βοήθεια τεσσάρων (4) γραμμών δεδομένων. Η τοποθέτηση των δεδομένων και η εκκίνηση του counter θα γίνονται με την βοήθεια δύο (2) γραμμών ελέγχου.

Εξηγείστε την λειτουργία του βασικού κυκλώματος του σχήματος 3 και του σχεδιασμένου από εσάς κυκλώματος.

Σχεδιάστε εδώ το κύκλωμα του μετρητή.

(B.2) Να σχεδιαστούν οι κυματομορφές της εξόδου Q2 για τις τιμές του πίνακα 1.

Clock	modulo N counter
1 KHz	16
	7
	6
	2

Πίνακας 1

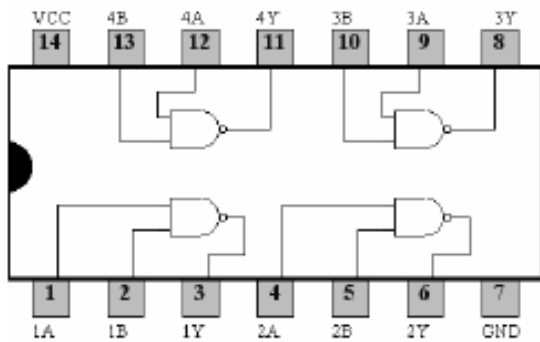
**** Σημείωση ****

Και στα δύο μέρη της άσκησης, οι έξοδοι των ΜΕΤΡΗΤΩΝ πρέπει να συνδέονται με Α.Μ.Α. (Αλφαριθμητική Μονάδα Απεικόνισης) για την οπτική παρουσίαση των αποτελεσμάτων.

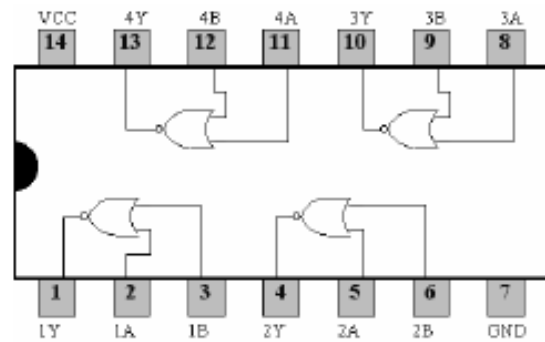
ΒΟΗΘΗΤΙΚΗ ΣΕΛΙΔΑ

ΠΑΡΑΡΤΗΜΑ

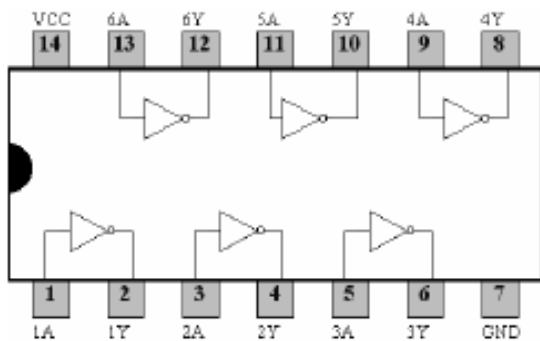
Διαγράμματα Ολοκληρωμένων



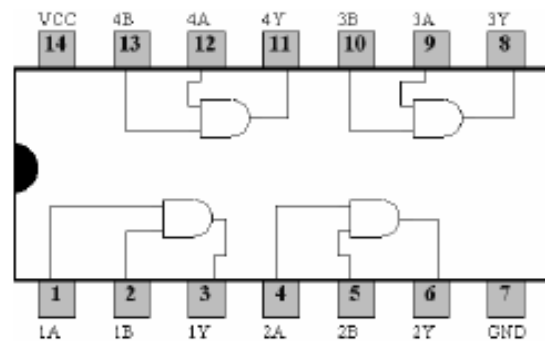
7400 Quad 2-Input NAND Gate



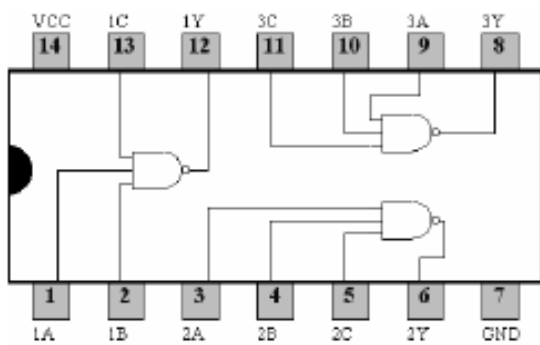
7402 Quad 2-Input NOR Gate



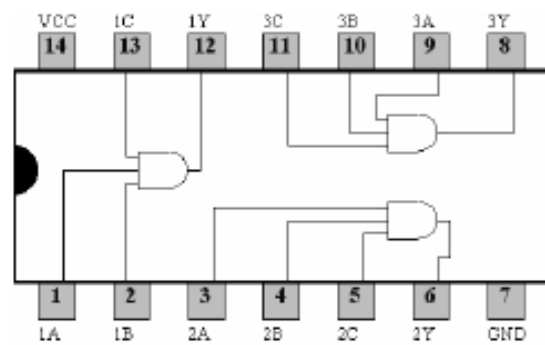
7404 Hex Inverter



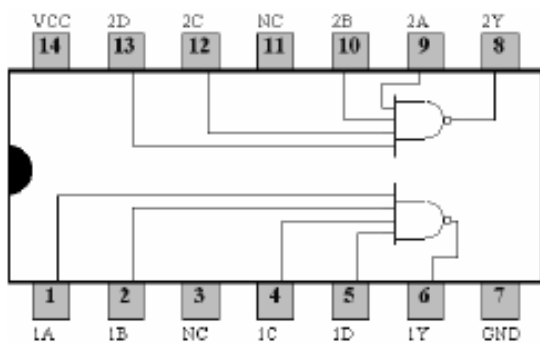
7408 Quad 2-Input AND Gate



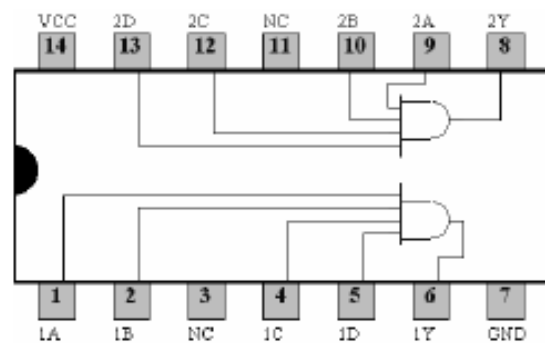
7410 Triple 3-Input NAND Gate



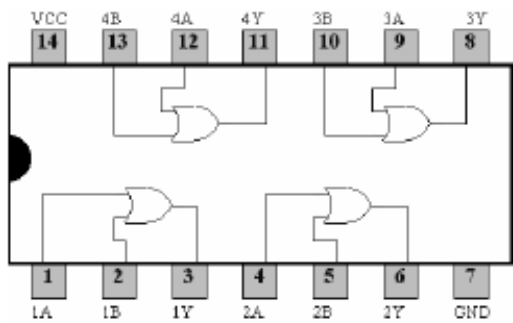
7411 Triple 3-Input AND Gate



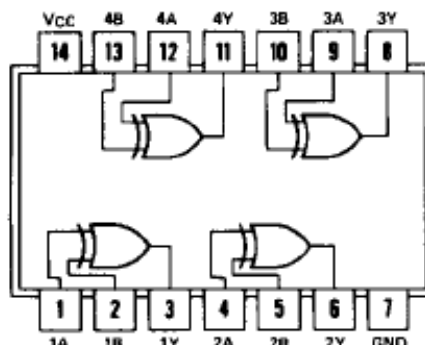
7420 Dual 4-Input NAND Gate



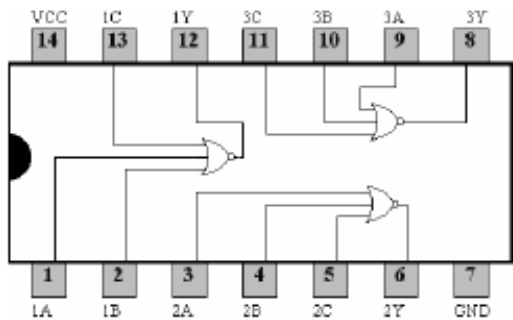
7421 Dual 4-Input AND Gate



7432 Quad 2-Input OR Gate



7486 Quad 2-Input XOR Gate

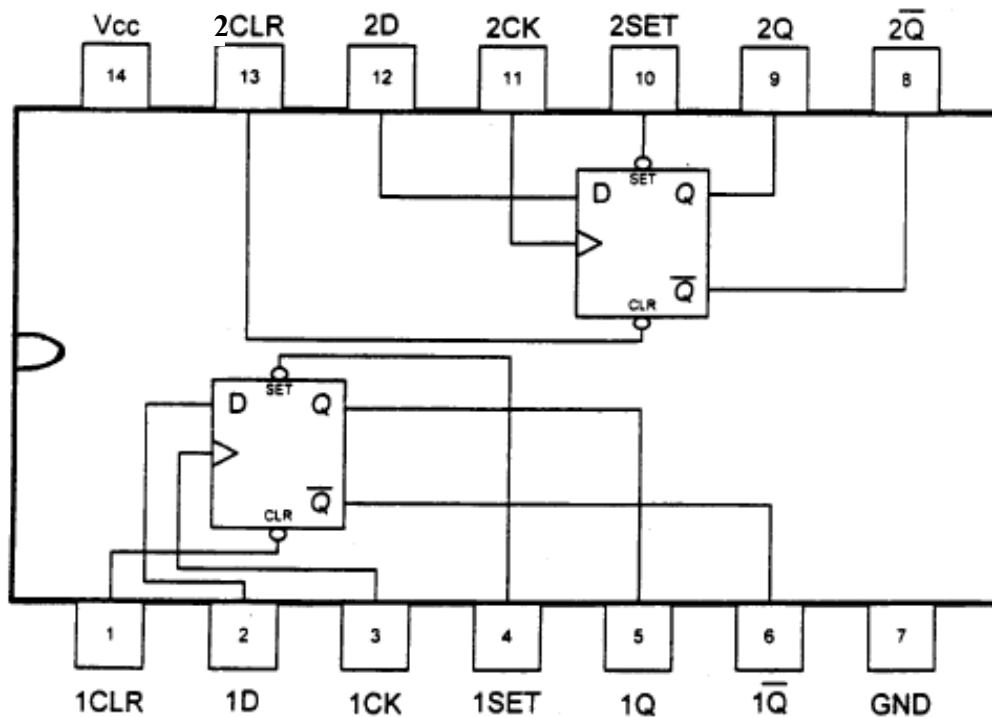


7427 Triple 3-Input NOR Gate

DUAL D-TYPE POSITIVE EDGE TRIGGERED FLIP-FLOPS WITH PRESET AND CLEAR

FUNCTION TABLE

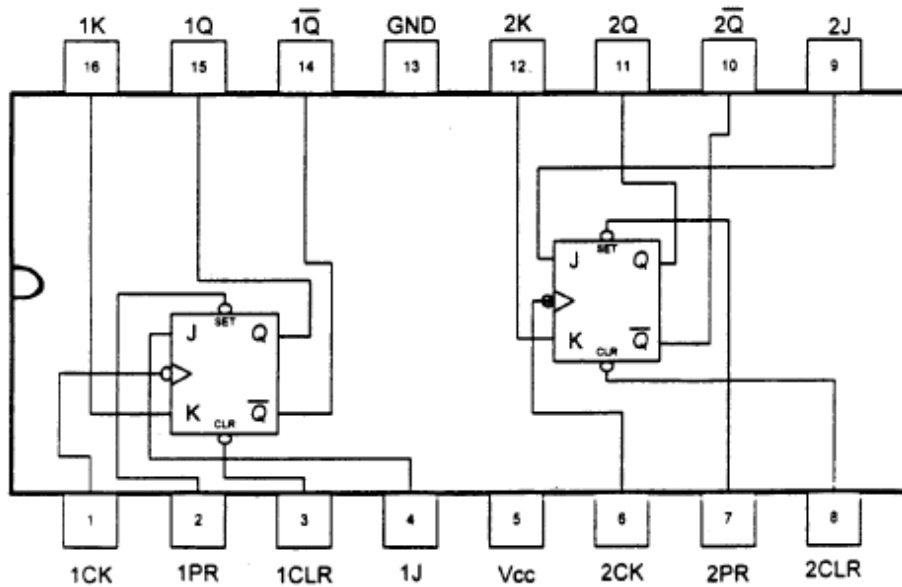
INPUTS				OUTPUTS	
PRESET	CLEAR	CLOCK	D	Q	Q'
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q ₀	Q' ₀



DUAL JK FLIP-FLOPS WITH PRESET AND CLEAR

FUNCTION TABLE

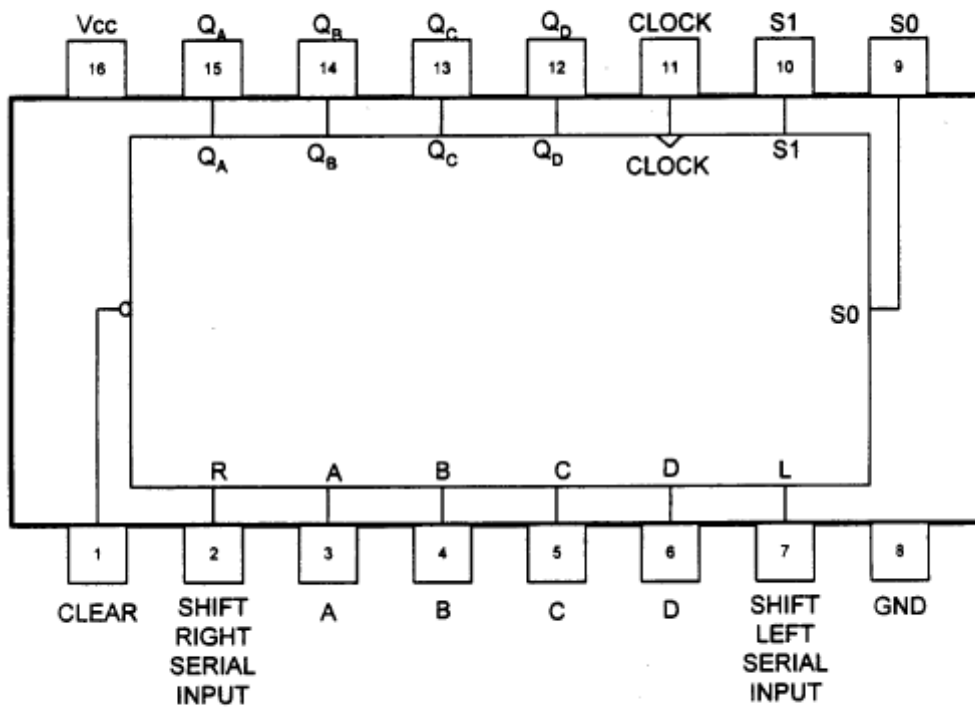
INPUTS					OUTPUTS	
PRESET	CLEAR	CLOCK	J	K	Q	Q'
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H	H
H	H	↓	L	L	Q ₀	Q' ₀
H	H	↓	H	L	H	L
H	H	↓	L	H	L	H
H	H	↓	H	H	TOGGLE	



4 - BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER

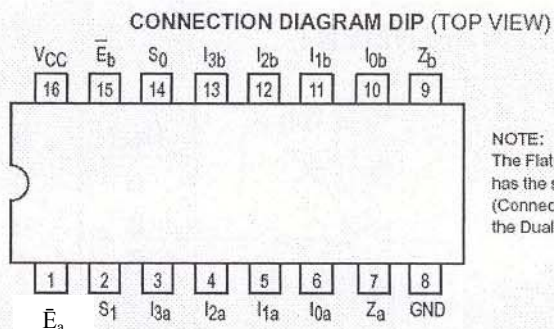
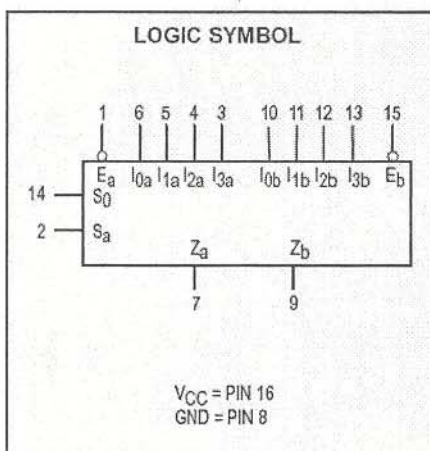
FUNCTION TABLE

CLEAR	MODE		CLOCK	INPUTS				OUTPUTS					
	S ₁	S ₀		SERIAL		PARALLEL				Q _A	Q _B	Q _C	Q _D
				LEFT	RIGHT	A	B	C	D				
L	X	X	X	X	X	X	X	X	X	L	L	L	L
H	X	X	L	X	X	X	X	X	X	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}
H	H	H	↑	X	X	a	b	c	d	a	b	c	d
H	L	H	↑	X	H	X	X	X	X	H	Q _{A0}	Q _{B0}	Q _{C0}
H	L	H	↑	X	L	X	X	X	X	L	Q _{A0}	Q _{B0}	Q _{C0}
H	H	L	↑	H	X	X	X	X	X	Q _{B0}	Q _{C0}	Q _{D0}	H
H	H	L	↑	L	X	X	X	X	X	Q _{B0}	Q _{C0}	Q _{D0}	L
H	L	L	X	X	X	X	X	X	X	Q _{A0}	Q _{B0}	Q _{C0}	Q _{D0}



74153

DUAL 4-INPUT MULTIPLEXER



NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package

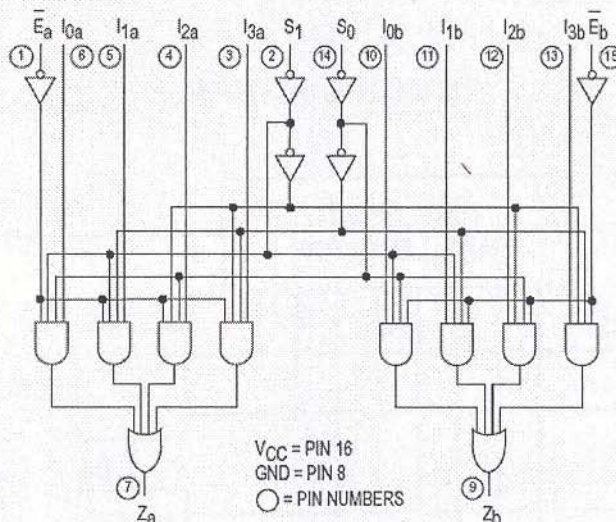
PIN NAMES

- S₀ Common Select Input
- E Enable (Active LOW) Input
- I₀, I₁ Multiplexer Inputs
- Z Multiplexer Output (Note b)

LOADING (Note a)

HIGH	LOW
0.5 U.L.	0.25
0.5 U.L.	0.25
0.5 U.L.	0.25
10 U.L.	5 (2.5)

LOGIC DIAGRAM



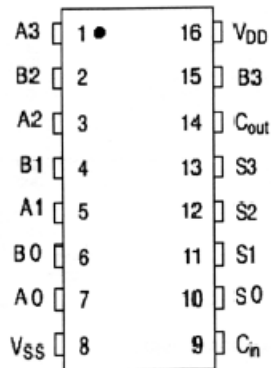
TRUTH TABLE

SELECT INPUTS		INPUTS (a or b)					OUTPUT
S ₀	S ₁	E	I ₀	I ₁	I ₂	I ₃	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

4008 4-BIT FULL ADDER

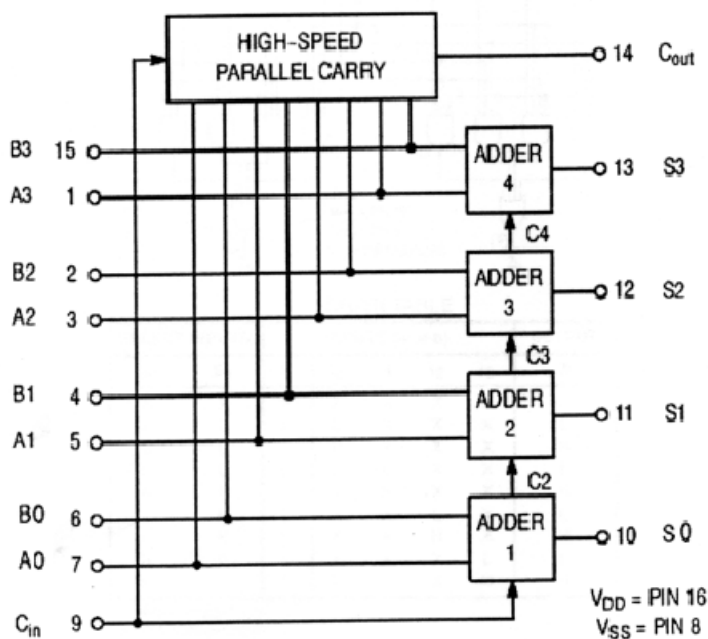
PIN ASSIGNMENT



TRUTH TABLE (One Stage)

C _{in}	B	A	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

BLOCK DIAGRAM



74LS181 – 4-bit ARITHMETIC LOGIC UNIT

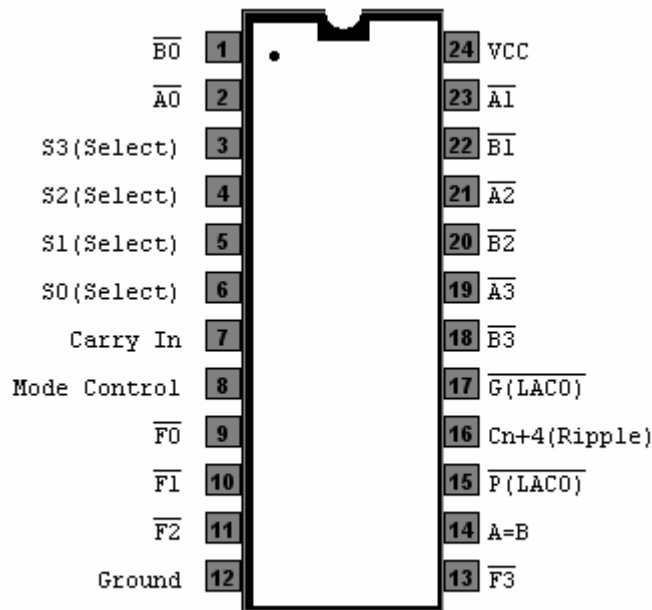


TABLE 1

SELECTION				ACTIVE-LOW DATA		
				M = H LOGIC FUNCTIONS	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0		C _n = L (no carry)	C _n = H (with carry)
L	L	L	L	$F = \overline{A}$	F = A MINUS 1	F = A
L	L	L	H	$F = \overline{AB}$	F = AB MINUS 1	F = AB
L	L	H	L	$F = \overline{A+B}$	F = \overline{AB} MINUS 1	F = \overline{AB}
L	L	H	H	F = 1	F = MINUS 1 (2's COMP)	F = ZERO
L	H	L	L	$F = \overline{A+B}$	F = A PLUS (A + \overline{B})	F = A PLUS (A + \overline{B}) PLUS 1
L	H	L	H	$F = \overline{B}$	F = AB PLUS (A + \overline{B})	F = AB PLUS (A + \overline{B}) PLUS 1
L	H	H	L	$F = A \oplus B$	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	$F = A + \overline{B}$	F = A + \overline{B}	F = (A + \overline{B}) PLUS 1
H	L	L	L	$F = \overline{AB}$	F = A PLUS (A + B)	F = A PLUS (A + B) PLUS 1
H	L	L	H	$F = A \oplus B$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = \overline{AB} PLUS (A + B)	F = \overline{AB} PLUS (A + B) PLUS 1
H	L	H	H	F = A + B	F = (A + B)	F = (A + B) PLUS 1
H	H	L	L	F = 0	F = A PLUS A [†]	F = A PLUS A PLUS 1
H	H	L	H	$F = \overline{AB}$	F = AB PLUS A	F = AB PLUS A PLUS 1
H	H	H	L	F = AB	F = \overline{AB} PLUS A	F = \overline{AB} PLUS A PLUS 1
H	H	H	H	F = A	F = A	F = A PLUS 1

[†]Each bit is shifted to the next more significant position.

TABLE 2

SELECTION				ACTIVE-HIGH DATA		
				M = H LOGIC FUNCTIONS	M = L; ARITHMETIC OPERATIONS	
S3	S2	S1	S0		C _n = H (no carry)	C _n = L (with carry)
L	L	L	L	$F = \overline{A}$	F = A	F = A PLUS 1
L	L	L	H	$F = \overline{A+B}$	F = A + B	F = (A + B) PLUS 1
L	L	H	L	$F = \overline{AB}$	F = A + \overline{B}	F = (A + \overline{B}) PLUS 1
L	L	H	H	F = 0	F = MINUS 1 (2's COMPL)	F = ZERO
L	H	L	L	$F = \overline{AB}$	F = A PLUS \overline{AB}	F = A PLUS \overline{AB} PLUS 1
L	H	L	H	$F = \overline{B}$	F = (A + B) PLUS \overline{AB}	F = (A + B) PLUS \overline{AB} PLUS 1
L	H	H	L	$F = A \oplus B$	F = A MINUS B MINUS 1	F = A MINUS B
L	H	H	H	$F = \overline{AB}$	F = \overline{AB} MINUS 1	F = \overline{AB}
H	L	L	L	$F = \overline{A+B}$	F = A PLUS AB	F = A PLUS AB PLUS 1
H	L	L	H	$F = A \oplus B$	F = A PLUS B	F = A PLUS B PLUS 1
H	L	H	L	F = B	F = (A + \overline{B}) PLUS AB	F = (A + \overline{B}) PLUS AB PLUS 1
H	L	H	H	F = AB	F = AB MINUS 1	F = AB
H	H	L	L	F = 1	F = A PLUS A [†]	F = A PLUS A PLUS 1
H	H	L	H	$F = A + \overline{B}$	F = (A + \overline{B}) PLUS A	F = (A + \overline{B}) PLUS A PLUS 1
H	H	H	L	F = A + B	F = (A + \overline{B}) PLUS A	F = (A + \overline{B}) PLUS A PLUS 1
H	H	H	H	F = A	F = A MINUS 1	F = A

[†]Each bit is shifted to the next more significant position.