ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΥΠΟΛΟΓΙΣΤΩΝ

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΣΤΗΜΗ ΤΟΥ ΗΛΕΚΡΟΛΟΓΟΥ ΜΗΧΑΝΙΚΟΥ

ΘΕΜΑ ΟΜΑΔΙΚΗΣ ΕΡΓΑΣΙΑΣ

**Ανάπτυξη και Μελέτη Λειτουργίας Αρχιτεκτονικών**

**Αθροιστών- Αφαιρετών**

ΟΜΑΔΑ 15

ΠΑΝΙΤΣΑΣ ΙΩΑΝΝΗΣ

ΠΑΝΤΕΛΗ ΧΡΙΣΤΟΦΟΡΟΣ

ΤΟΦΑΛΙΔΗΣ ΝΕΚΤΑΡΙΟΣ

ΑΡΕΣΤΗ ΑΛΕΞΑΝΔΡΟΣ

ΣΙΑΛΗΣ ΧΡΙΣΤΟΔΟΥΛΟΣ

ΜΑΡΚΟΥ ΑΝΔΡΕΑΣ

Επιβλέπων Καθηγητής

Θεοδωρίδης Γεώργιος

**ΠΕΡΙΕΧΟΜΕΝΑ**

0.0) Εισαγωγή στην έννοια του αθροιστή-------------------------------------- σελίδα 3

1.0) Ripple Carry Adder-------------------------------------------------------------- σελίδα 4

1.1) Δομή και Λειτουργία FA------------------------------------------------------- σελίδα 4-6

1.2 )Κυκλώματα 4 και 8 bit για τον RCA----------------------------------------- σελίδα 6-7

1.3) Πλεονεκτήματα και μειονεκτήματα RCA--------------------------------- σελίδα 8

1.4) Καθυστέρηση(delay)του RCA------------------------------------------------ σελίδα 9-10

2.0) Carry Look Ahead Adder------------------------------------------------------ σελίδα 10-11

2.1) Πλεονεκτήματα και Μειονεκτήματα CLA--------------------------------- σελίδα 11

2.2) Καθυστέρηση(delay) του CLA----------------------------------------------- σελίδα 12-13

2.3) Ισοδύναμο του Propagate και του Genarate---------------------------- σελίδα 13-14

3.0) Carry Skip Adder----------------------------------------------------------------- σελίδα 15-16

3.1)Πλεονεκτήματα και Μειονεκτήματα CSA---------------------------------- σελίδα 16

3.2)Καθυστέρηση(delay) του CSA------------------------------------------------- σελίδα 16-17

4.0) Carry Select Adder--------------------------------------------------------------- σελίδα 18-21

4.1)Πλεονεκτήματα και Μειονεκτήματα CSA---------------------------------- σελίδα 21

4.2)Καθυστέρηση(delay) του CSA------------------------------------------------- σελίδα 21

5.0)Λογισμικό DEEDS----------------------------------------------------------------- σελίδα 22-24

6.0)Βιβλιογραφία---------------------------------------------------------------------- σελίδα 25

**0.0)ΕΙΣΑΓΩΓΗ**

Το κύκλωμα της πρόσθεσης είναι το βασικότερο κύκλωμα για την εκτέλεση των αριθμητικών πράξεων ενός ψηφιακού συστήματος καθώς είναι η βάση για την κυκλωματική υλοποίηση και των υπολοίπων αριθμητικών πράξεων. Για το σκοπό αυτό έχει προταθεί ένας μεγάλος αριθμός εναλλακτικών αρχιτεκτονικών. Το βασικό χαρακτηριστικό κάθε αρχιτεκτονικής είναι η αποδοτική μετάδοση των κρατούμενων μεταξύ των ψηφίων.   
  
Στην παρούσα εργασία αρχικά θα μελετηθεί ένα σύνολο από αρχιτεκτονικές άθροισης (πχ. Carry skip, carry select,carry lookahead) και θα κατανοηθεί η λειτουργία τους. Στη συνέχεια θα γίνει η υλοποίηση τους με κατάλληλο εργαλείου σχεδιασμού στον υπολογιστή και η εκτέλεση των απαραίτητων προσομοιώσεων για την επιβεβαίωση της ορθής λειτουργίας των υλοποιήσεων(DEEDS).   
  
Οι γνώσεις που απαιτούνται είναι ο σχεδιασμός συνδυαστικών ψηφιακών κυκλωμάτων που καλύπτεται από την ύλη του μαθήματος «Εισαγωγή στη Ψηφιακή Λογική Σχεδίαση. Στόχος της συγκεκριμένης εργασίας είναι η υλοποίηση 4 αθροιστών των 12 bit o καθένας οι οποίοι πρέπει να κινούνται σε δύο βασικούς άξονες.Οι δύο αυτοί άξονες θα αναφερθούν αναλυτικά πιο κάτω.Oι αθροιστές που υλοποιήσαμε είναι οι εξής: Ripple Carry Adder(RCA) , Carry Skip Adder(CSA), Carry Lookahead Adder(CLA) και Carry Select Adder(CSA).Στη συνέχεια, θα αναφερθούν αναλυτικά τα χαρακτηριστικά κάθε αθροιστή, οι καθυστερήσεις που βάζουν οι λογικές πύλες, τα διαγράμματα καθυστέρησης καθώς και τα αρνητικά κάθε αθροιστή.

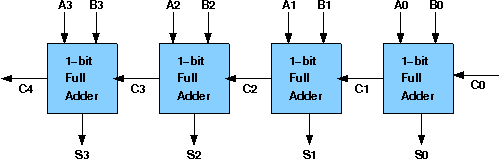
Επί της ουσίας, ο αθροιστής γενικότερα είναι συνήθως ένα ψηφιακό κύκλωμα που βασικός στόχος του είναι η πρόσθεση δυαδικών αριθμών. Οι αθροιστές χρησιμοποιούνται ευρέως σε υπολογιστικά συστήματα οι οποίοι αποτελούνται συνήθως από κυκλώματα ημιθροιστών , πλήρων αθροιστών και λογικών πυλών.

O δυαδικός αθροιστής είναι ένα ψηφιακό κύκλωμα που παράγει το αριθμητικό άθροισμα δυο δυαδικών ψηφίων. Μπορεί να κατασκευαστεί με πλήρεις αθροιστές, εάν αυτοί συνδεθούν σε σειρά και το κρατούμενο εξόδου του κάθε πλήρους αθροιστή συνδεθεί με το κρατούμενο εισόδου του επόμενου πλήρους αθροιστή.Η άθροιση αριθμών με n bit απαιτεί μια <<αλυσίδα>> αποτελούμενη από n πλήρεις αθροιστές, ή μια αλυσίδα αποτελούμενη από ένα ημιαθροιστή και n-1 πλήρεις αθροιστές.

**1.0) ΑΘΡΟΙΣΤΗΣ ΡΙΠΗΣ ΚΡΑΤΟΥΜΕΝΟΥ (RIPPLE CARRY ADDER)**

Ο ripple carry adder είναι ο πιο απλός αθροιστής σε σχέση με τους υπόλοιπους αθροιστές στην υλοποίησή του. Αποτελείται από πλήρεις αθροιστές(Full Adders) oι οποίοι είναι συνδεδεμένοι σε σειρά. Ο αριθμός των πλήρων αθροιστών που θα χρησιμοποιήσουμε καθορίζει και τα bit του αριθμού που υπολογίζουμε. Για παράδειγμα ένας αθροιστής ριπής κρατουμένου που αποτελείται από 4 πλήρεις αθροιστές είναι ένας 4 bit RCA ο οποίος προσθέτει δύο αριθμούς 4 ψηφίων.

Σχηματικά μια εικόνα του ενός 4bit RCA είναι:



**Σχήμα 1.1: 4 bit Ripple Carry Adder**

Έστω ότι θέλουμε να προσθέσουμε δύο δυαδικούς αριθμούς τους Α3Α2Α1Α0=0101 και Β3Β2Β1Β0=0001

Θεωρούμε ότι το LSB των αριθμών είναι το Α0 και το Β0.

Κάθε FA δέχεται 3 εισόδους, εκ των οποίων οι 2 είσοδοι είναι τα bit του αριθμού και η τρίτη είσοδος είναι το κρατούμενο εισόδου. Η έξοδος κάθε FA είναι το bit του τελικού αποτελέσματος ενώ το τελικό αποτέλεσμα είναι το S3S2S1S0.

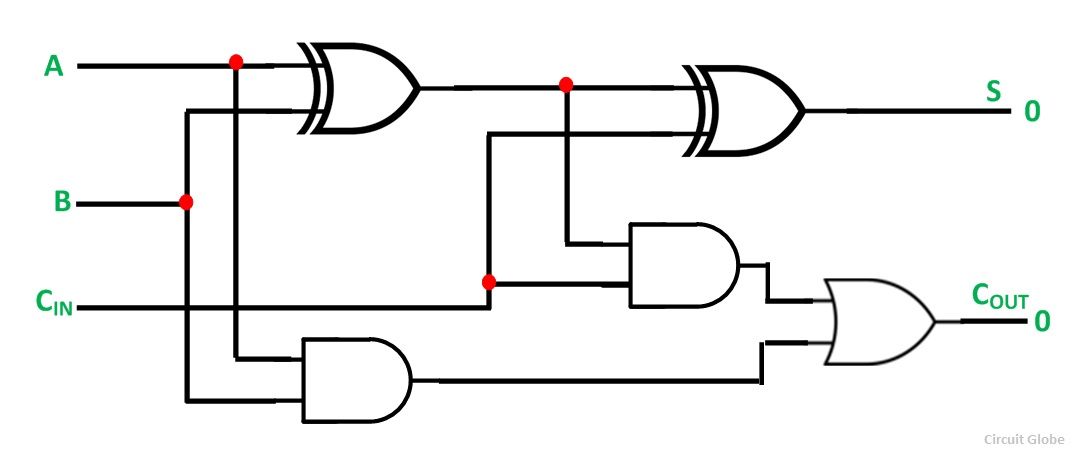
Σε αυτό το σημείο θα παρουσιάσουμε τη δομή του FA .

**1.1)Δομή και Λειτουργία FA**

O πλήρης αθροιστής είναι ένα συνδυαστικό κύκλωμα που παράγει το αριθμητικό άθροισμα τριών bit.Το κύκλωμα αυτό έχει τρείς εισόδους και δύο εξόδους. Δύο από τις μεταβλητές εισόδου ,Α και Β, αναπαριστούν τα δυο bit της ίδιας τάξης που θα προστεθούν. Η τρίτη είσοδος Cin αναπαριστά το κρατούμενο από την προηγούμενη αμέσως λιγότερο σημαντική θέση. Απαιτείται η χρήση δύο εξόδων ,επειδή το αριθμητικό άθροισμα τριών δυαδικών ψηφίων που μπορούν να λάβουν τιμές από 0 έως 3 μπορεί να έχει ως αποτέλεσμα το δυαδικό 2 ή το δυαδικό 3,τα οποία χρειάζονται 2 bit για την αναπαράστασή τους. Οι δυο αυτοί έξοδοι ονομάζονται S

και Cout. H δυαδική μεταβλητή S δίνει την τιμή του λιγότερου σημαντικού bit ενώ η δυαδική μεταβλητή Cout δίνει το κρατούμενου εξόδου.

Το κύκλωμα που αποτελεί το FA είναι το παρακάτω:



**Σχήμα 1.2: Kύκλωμα Full Adder**

Στη ουσία ένας πλήρης αθροιστής είναι δύο ημιαθροιστές και μία πύλη ΟR.Το κρατούμενο εξόδου κάθε FA στο RCA το συνδέουμε στο κρατούμενο εισόδου του επόμενου αθροιστή εκτός από το πρώτο και το τελευταίο αθροιστή. Στο τελευταίο αθροιστής το κρατούμενο εξόδου είναι το MSB ψηφίο του τελικού αριθμού. O πίνακας αληθείας του Full Adder είναι ο παρακάτω:



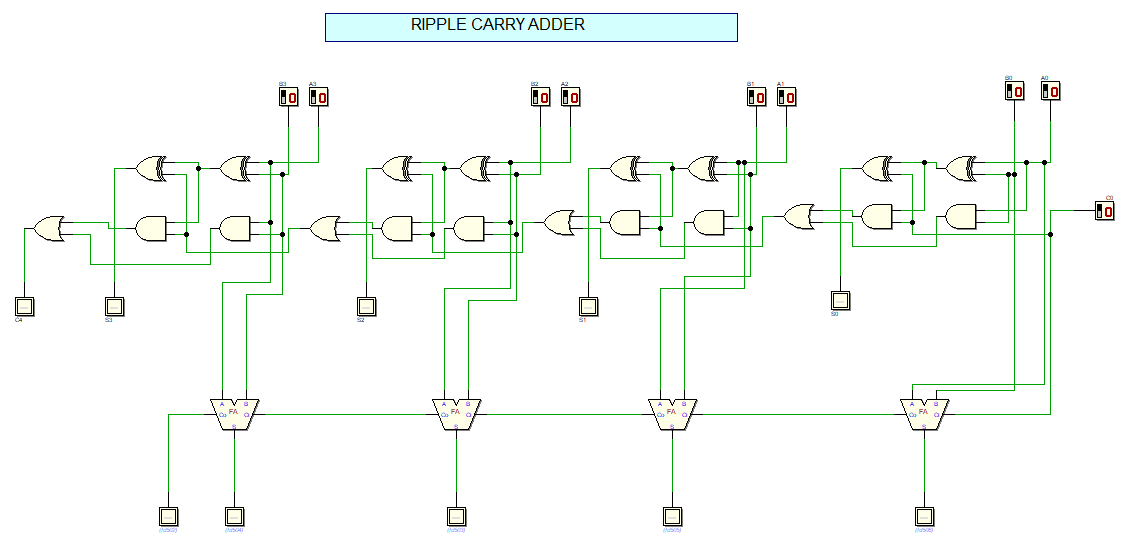
**Σχημα 1.3: Πίνακας Αληθείας Πλήρους Αθροιστή**

Μέσω του χάρτη Καρνώ προκύπτουν οι εξισώσεις που καθορίζουν το Sum και το Cout.Πιο συγκεκριμένα ,η έξοδος sum θα ισούται με S=(A XOR B)XOR C ενώ το Cout θα ισούται με C=AB+(A XOR B)Ci.

Aφού αναλύσαμε το κύκλωμα του Full Adder πλέον μπορούμε να προχωρήσουμε στο RCA.

**1.2) Κυκλώματα 4 και 8 bit Ripple Carry Adder**

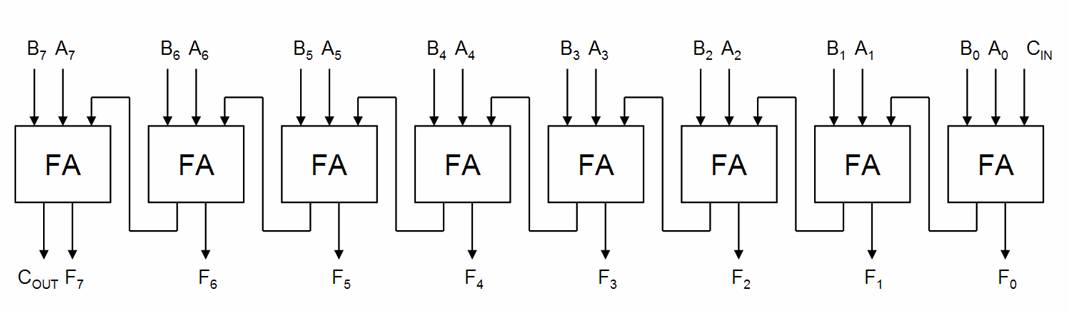
Παρακάτω φαίνεται ένας 4bits RCA ο οποίος αποτελείται από τους 4 FA σε σειρά:



**Σχήμα 1.4: Ψηφιακό κύκλωμα 4 bit Ripple Carry Adder μέσω του προγράμματος DEEDS**

Στη συγκεκριμένη εργασία ωστόσο στόχος είναι η υλοποίηση 12 bit αθροιστών .Όμως, με την σχεδίαση 4bit Block μας δίνεται η δυνατότητα να σχεδιάσουμε αθροιστές με οποιοδήποτε πολλαπλάσιο του 4.Για παράδειγμα, αν θέλουμε να σχεδιάσουμε ένα αθροιστή των 16 bits θα χρησιμοποιήσουμε 4 block των 4 bits ή δύο block των 8bits.

Aν θέλουμε να κατασκευάσουμε ένα RCA tων 8 bits θα χρειαστούμε 8 FA συνδεδεμένους στη σειρά.H παρακάτω εικόνα δείχνει ένα 8 bit RCA.

**Σχήμα 1.5: RCA των 8 bits**

Η υλοποίηση ενός 4 bit RCA και ενός 8 bit RCA όπως παρατηρούμε δεν αλλάζει. Με την ίδια λογική κατασκευάζεται ένας n bit RCA.

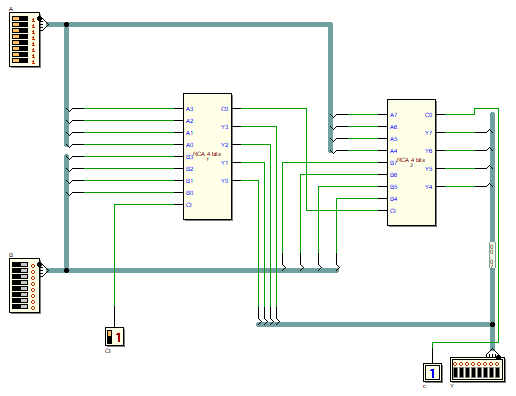
**1.3)ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΚΑΙ ΜΕΙΟNΕΚΤΗΜΑΤΑ RCA**

Ο αθροιστής ριπής κρατουμένου αποτελεί ένα αθροιστή ο οποίος είναι εύκολος στην υλοποίησή του και για αυτό το λόγο χρησιμοποιείται σε αρκετά ψηφιακά κυκλώματα. Eπίσης , το κόστος κατασκευής του είναι μικρό λόγω της μικρής πολυπλοκότητας του. Το βασικό πρόβλημα του RCA είναι η καθυστέρηση για την εμφάνιση του Cout, δηλαδή η καθυστέρηση για τη διάδοση του κρατουμένου Cout. O χρόνος διάδοσης των κρατουμένων είναι σημαντικό χαρακτηριστικό του αθροιστή, επειδή περιορίζει την ταχύτητα με την οποία θα προστεθούν οι δυο αριθμοί. Κατά την πρόσθεση δύο δυαδικών αριθμών εν παραλλήλω ,απαιτείται όλα τα bit τόσου του πρώτου όσο του δεύτερου προσθετέου να είναι ταυτόχρονα διαθέσιμα. Όμως όπως ισχύει σε οποιοδήποτε συνδυαστικό κύκλωμα όλα τα αντίστοιχα σήματα πρέπει να προλάβουν να διαδοθούν μέσω των πυλών, πριν εμφανιστεί στην έξοδο το σωστό άθροισμα. O συνολικός χρόνος διάδοσης ισούται με το γινόμενο της καθυστέρησης διάδοσης μιας τυπικής πύλης επί τον αριθμό των επιπέδων πυλών του κυκλώματος. Η τιμή του τελικού κρατούμενο εξόδου εξαρτάται από τον σωστό υπολογισμό των προηγουμένων κρατούμενων. Συνεπώς, μόνο μετά από τη σωστή διάδοση των κρατούμενων μέσα από όλα τας στάδια του αθροιστή και τη χρήση τους στους αντίστοιχους υπολογισμούς, η τελευταία έξοδος και το κρατούμενο θα καταλήξουν στη σωστή τελική τιμή τους. Ο αριθμός των επιπέδων πυλών που πρέπει να λειτουργήσουν για να γίνει η διάδοση των κρατούμενων μπορεί να βρεθεί από το κύκλωμα το πλήρους αθροιστή. Σε ένα RCA των n bit υπάρχουν n πλήρεις αθροιστές, συνεπώς το κρατούμενο εξόδου υπολογίζεται από 2n επίπεδα πυλών,τα οποία παρεμβάλλονται μεταξύ του Cin και του Cout.

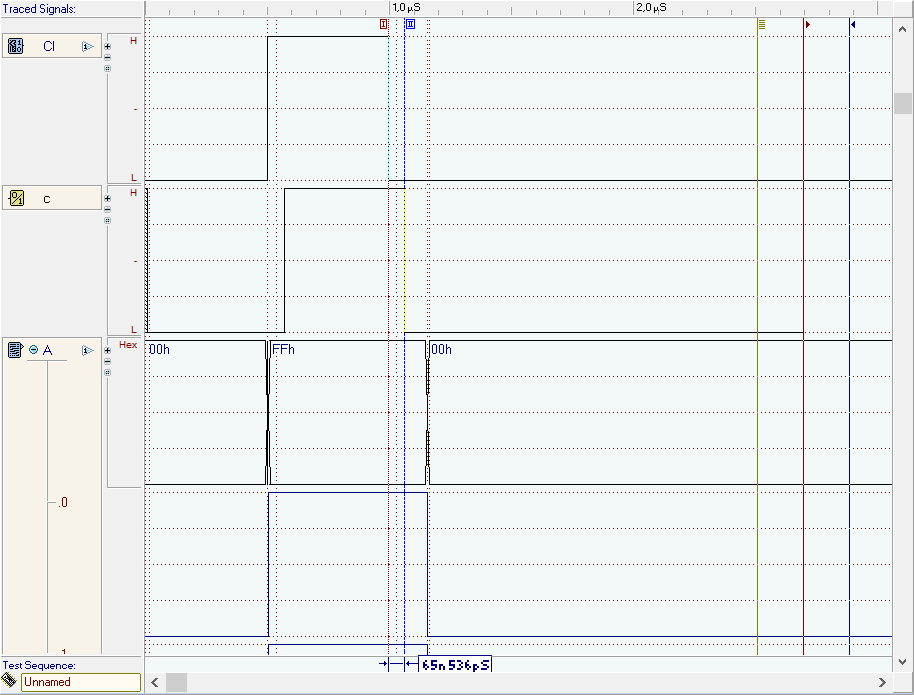
Αν ο αθροιστής είναι των n bits τότε η καθυστέρηση για τον υπολογισμό του Cout και συνεπώς του αριθμού που μας ενδιαφέρει είναι n x ta όπου ta είναι η καθυστέρηση που βάζει ένας FA για τον υπολογισμό του κρατούμενου εξόδου. Συνεπώς παρατηρούμε ότι η καθυστέρηση στον RCA είναι γραμμική. Η γραμμική καθυστέρηση είναι ένα σημαντικό πρόβλημα για την ταχύτητα του κυκλώματος. Γενικότερα, στα ψηφιακά κυκλώματα επιδιώκουμε την μέγιστη ταχύτητα με το ελάχιστο κόστος και αυτό γιατί όλες οι αριθμητικές πράξεις υλοποιούνται μέσω διαδοχικών προσθέσεων .Όπως καταλαβαίνουμε , ο χρόνος που απαιτείται για να ολοκληρωθεί μια διαδικασία είναι κρίσιμος.

**1.3)KAΘΥΣΤΕΡΗΣΗ Ripple Carry Adder**

Όπως ,αναφέραμε και πριν κάθε αθροιστής που υλοποιείται για την εκτέλεση της άθροισης μεταξύ δύο αριθμών εισάγει μία καθυστέρηση για να υπολογίσει το άθροισμα τους. Μέσω του προγράμματος DEEDS στο οποίο θα αναφερθούμε αναλυτικά στη συνέχεια είναι δυνατή η μέτρηση της καθυστέρησης που εισάγει ο αθροιστής .Έστω ότι έχουμε ένα 8 bit αθροιστή και θέλουμε να υπολογίσουμε το άθροισμα των αριθμών 00000000 και 11111111 με κρατούμενο εισόδου 1.Θεωρητικά , το αποτέλεσμα αυτής της αριθμητικής πράξης είναι 100000000.Με τη βοήθεια του προγράμματος DEEDS και σχεδιάζοντας το αθροιστή από την αρχή βρήκαμε τα εξής αποτελέσματα Υ=100000000 και delay=65ns τα οποία φαίνονται και στις παρακάτω εικόνες.



**Σχήμα 1.6 Υλοποίηση 8 bit RCA μέσω του προγράμματος DEEDS .Πρόσθεση των αριθμών 111111111 και 00000000 με κρατούμενο εισόδου 1.**

****

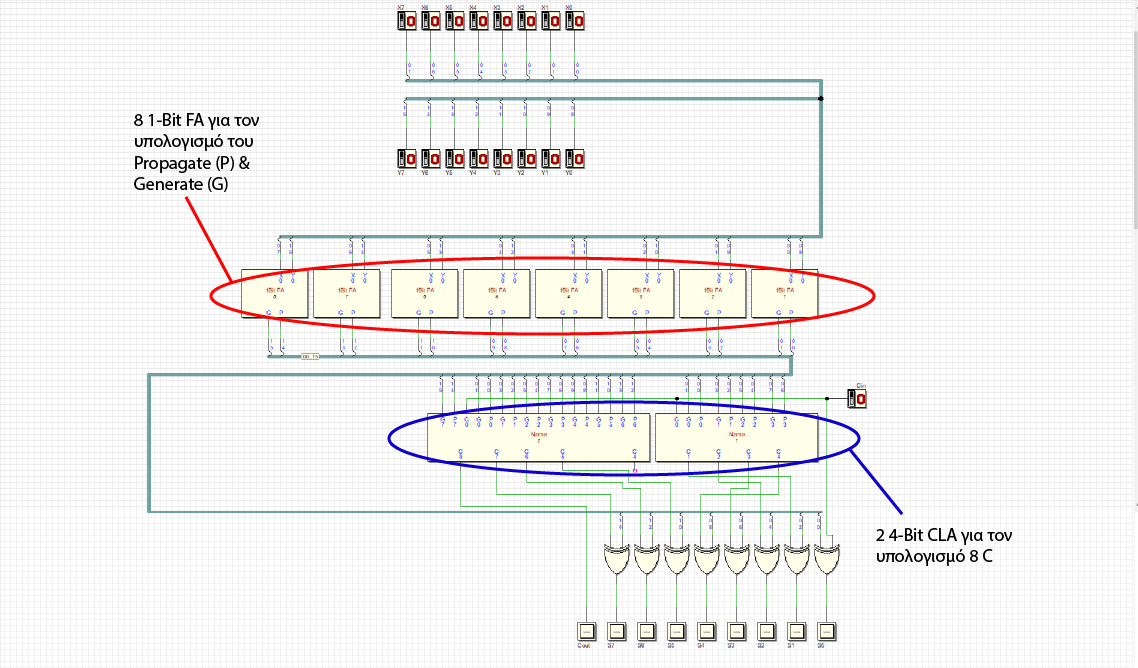
**Σχήμα 1.7 Μέτρηση Καθυστέρησης 8 bit RCA.O τετραγωνικός παλμός δηλώνει το λογικό ΄΄1΄΄.Το c παριστάνει το κρατούμενο εξόδου και το CI το κρατούμενο εισόδου. Η καθυστέρηση διάδοσης του κρατούμενου είναι 65 ns.**

Η καθυστέρηση στον RCA δίνεται από τον γραμμική σχέση:

**TCRA=tXOR  + 2(n-1)tAND**

**2. 0)Carry Look Ahead Adder:**

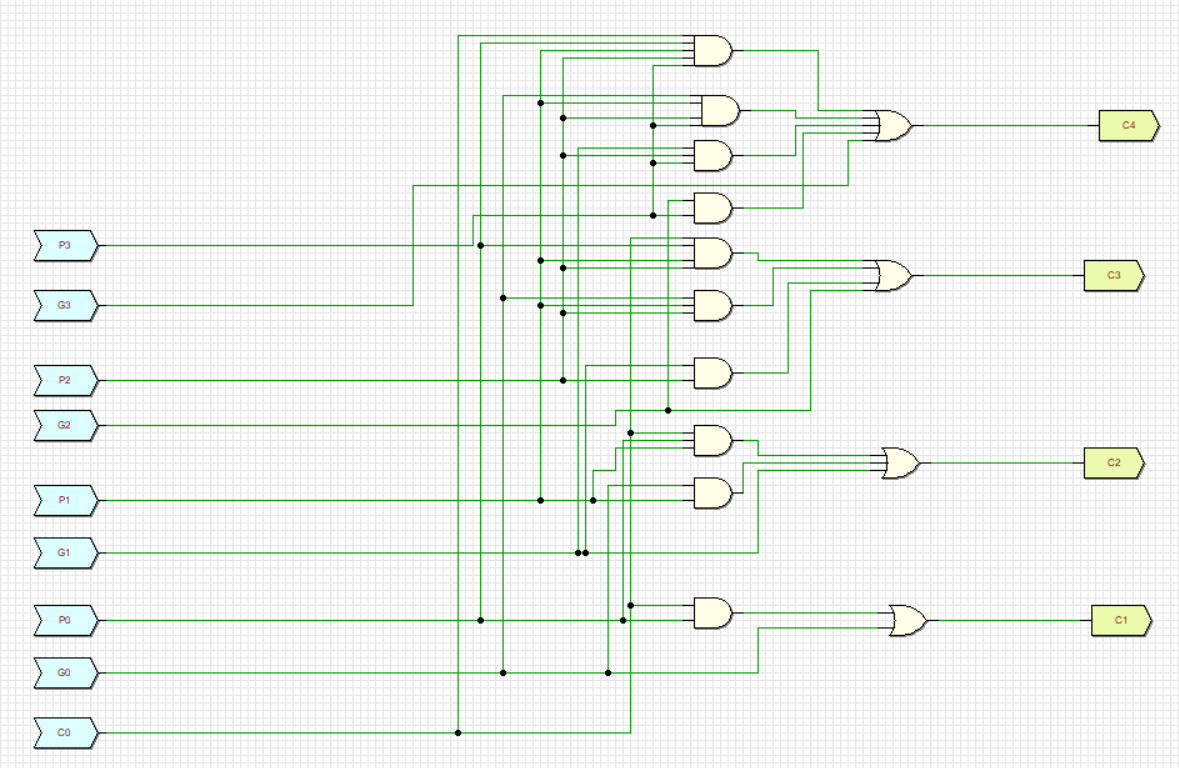
Ένας αθροιστής carry-lookahead (CLA) είναι ένας τύπος αθροιστή που χρησιμοποιείται στην ψηφιακή λογική. Ο αθροιστής CLA βελτιώνει την ταχύτητα μειώνοντας το χρόνο το χρόνο που απαιτείται για τον προσδιορισμό των bits μεταφοράς. Μπορεί να αντιπαραβληθεί με τον απλούστερο , αλλά συνήθως πιο αργό, Ripple-Carry Adder (RCA), για τον οποίο το bit μεταφοράς υπολογίζεται παράλληλα με το άθροισμα bit και κάθε bit πρέπει να περιμένει μέχρι να υπολογιστεί το προηγούμενο bit μεταφοράς για να αρχίσει ο υπολογισμός του δικού του αποτελέσματος και Сarry Bits. Ο αθροιστής Carry Look Ahead υπολογίζει ένα ή περισσότερα ψηφία μεταφοράς πριν από το άθροισμα, πράγμα που μειώνει τον χρόνο αναμονής για τον υπολογισμό του αποτελέσματος των δυαδικών ψηφίων μεγάλης τιμής του αθροιστή.



**Σχήμα 2.1 8 bit Κύκλωμα με CLA**

**2. 1 Πλεονεκτήματα και Μειονεκτήματα του CLA:**

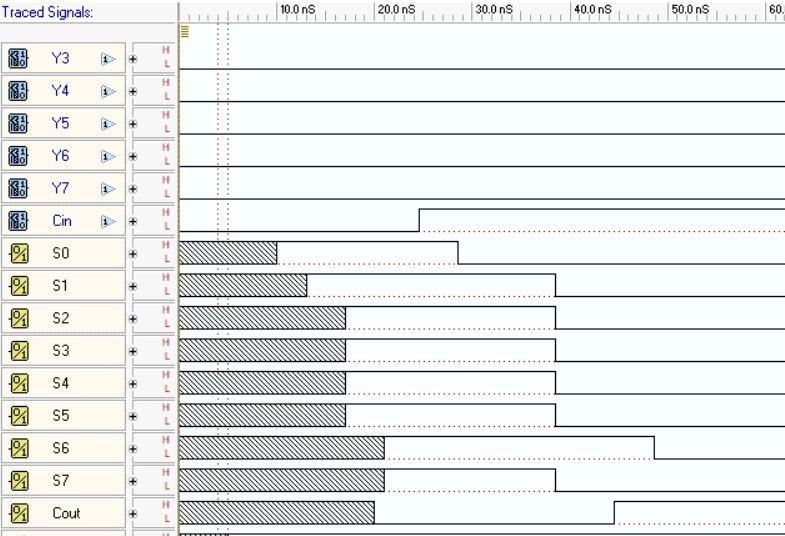
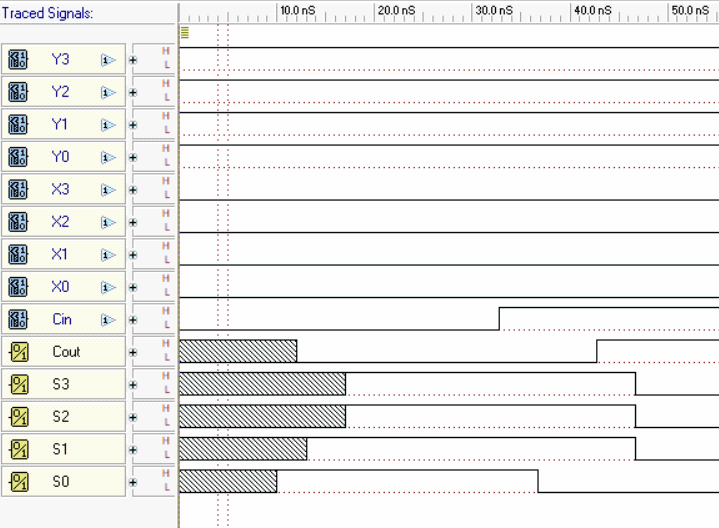
1. Δεν χρειάζεται να περιμένουμε την διάδοση των Carries ( C ) για να πάρουμε το άθροισμα ( S / Sum )
2. Εάν κατασκευάσουμε το κύκλωμα εντελώς από πύλες 2 εισόδων AND ή 2 εισόδων OR , τότε το καλύτερο που μπορούμε να κάνουμε είναι το O (log N) όπου N είναι ο αριθμός των bits στην προσθήκη. Το οποίο είναι πιο αποδοτικό από το Ο ( Ν ).
3. Το carry logic block αρχίζει και γίνεται περίπλοκο όσο αυξάνονται τα bits
4. Μειώνεται η αποδοτικότητα του όσο αυξάνονται τα bits



**Σχήμα 2.2 4-bit CLA**

**2.2)Καθυστέρηση Carry Look Ahead Adder(CLA)**

Η αποδοτικότητα του CLA εξαρτάται απο τον αριθμό των bits που θα χρησιμοποιήσουμε. *Για παράδειγμα* αν θέλουμε να προσθέσουμε 2 8-bit, θα πρέπει να προβλέψουμε 8 C. Όπως βλέπουμε στο Σχ.2 όσο ανεβαίνει ο αριθμός των С γίνεται πιο περίπλοκο και πέφτει η αποδοτικότητα του CLA διότι για την πρόβλεψη του κάθε C αυξάνονται και οι λογικές πύλες που πρέπει να χρησιμοποιήσουμε.



**Σχήμα 2.3 4 Bit CLA ( Delay = 12ns )** **Σχήμα 2.4 8 Bit CLA ( Delay=24ns )**

**Γράφημα 2.5 Delays 4 and 8 Bits CLA**

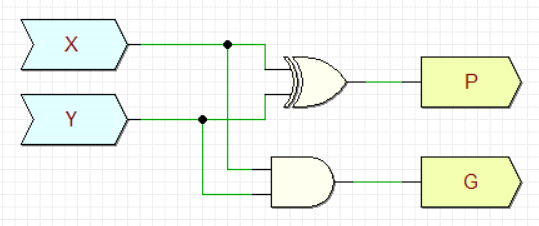
Η καθυστέρηση στον Carry Look Ahead Adder υπολογίζεται από τον τύπο:



Όπου m είναι ο αριθμός των bits του block και n το πλήθος των bits του αρχικού αριθμού.

**2.3) Ισοδύναμο του Propagate και του Genarate**

* Τον υπολογισμό του Propagate ( P ) που προκύπτει απο την εξίσωση Pn = Xn ⊕ Yn
* Τον υπολογισμό του Generate ( G ) που προκύπτει απο την εξίσωση Gn = XnYn



**Σχήμα 2.6: Υπολογισμός Propagate and Genarate**

***Yπολογισμός των κρατούμενων συναρτήσει του κρατούμενου εισόδου***

Υπολογίζεται απο την εξίσωση Cn = Gn-1 + Pn-1Cn-1 και αντικαθιστώντας τα C στις εξισώσεις:

Για παράδειγμα αν θέλουμε να υπολογίσουμε ένα 8-bit :

* C1 = G0 + P0C0
* C2 = G1 + P1C1  --> C2 = G1 + P1(G0 + P0C0) --> C2 = G1 + P1G0 + P1P0C0
* C3 = G2 + P2C2 --> C3 = G2 + P2(G1 + P1G0 + P1P0C0) --> C3 = G2 + P2G1 + P2P1G0 + P2P1P0C0

**...**

* C8 = G7 + P7C7

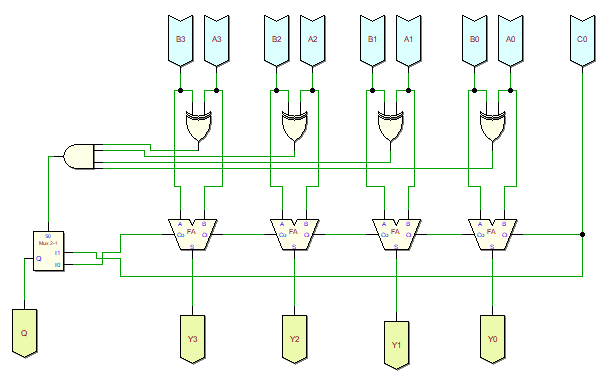
Όπως βλέπεται για κάθε C αυξάνονται τα στοιχεία τους, άρα περισσότερες λογικές πύλες με συνέπεια να μειώνεται η αποδοτικότητα του Carry Look Ahead .

Αφού υπολογιστούν τα C μπορούμε να υπολογίσουμε τα sum bits ( S ) με την εξίσωση:

* S0 = C0 + P0
* S1 = C1 + P1
* Sn = Cn + Pn

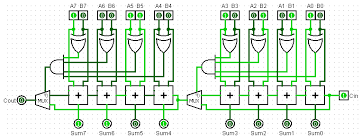
**3.0)Carry Skip Adder(CSA)**

O carry skip adder ή carry bypass adder(αθροιστής μεταφοράς μέσω παράκαμψης) είναι ένας αθροιστής ο οποίος βελτιώνει την καθυστέρηση που εισάγει o Ripple Carry Adder.Καταλαμβάνει μεγαλύτερη έκταση από τον RCA επειδή το κύκλωμα του περιέχει περισσότερες πύλες και άλλα ψηφιακά στοιχεία όπως ο πολυπλέκτης. Σε αυτό το σημείο πρέπει να τονίσουμε ότι έχουμε αυξημένη πολυπλοκότητα (λόγω του αριθμού πυλών) ώστε να μειωθεί ο συνολικός χρόνος καθυστέρησης της διάδοσης των κρατούμενων. Η δομή ενός 4 bit CSA είναι η παρακάτω:



**Σχήμα 3.1: Carry Skip Adder 4 bit**

Η βασική ιδέα του CSA στηρίζεται στο εξής: Έστω ότι έχουμε ένα 8 bit CRS και θέλουμε να υπολογίσουμε το άθροισμα των αριθμών 00000000 και 11111111 με κρατούμενο εισόδου 1 όπως είχαμε κάνει και στον RCA. Ο 8 bit CSA δομείται από 2 block των 4 bit όπως στο σχήμα 3.1.Αρχικά ,ο αθροιστής θα υπολογίσει τα propagate των 4 αρχικών ψηφίων(από το LSB έως το τέταρτο λιγότερο σημαντικό ψηφίο).Εάν και τα τέσσερα propagate είναι ίσα με 1 τότε απευθείας το Cout θα ισούται με το Cin,αφού θα ενεργοποιηθεί η είσοδος Ι1 του ΜUX.Αν τουλάχιστον ,ένα από τα propagate δεν είναι ίσο με ΄1΄ τότε θα ενεργοποιηθεί η είσοδος Ι0 του ΜUX .Σε αυτή την περίπτωση ο carry skip adder θα περιμένει την διάδοση του κρατουμένου μέσα από τους τέσσερις FA.Στο παρακάτω σχήμα φαίνονται οι συνδέσεις εσωτερικό ενός 8 bit CSA.



**Σχήμα 3.2: Carry skip 8 bit**

**3.1)Πλεονεκτήματα - Μειονεκτήματα CSA.**

Όπως αναφέρθηκε και στην αρχή, πρωταρχικός στόχος του CSA είναι η μείωση της καθυστέρησης της διάδοσης του κρατουμένου .Αυτό αποτελεί και το βασικό του πλεονέκτημα σε σχέση με τους υπόλοιπους αθροιστές. Από μετρήσεις μας οι οποίες θα παρουσιαστούν στο τέλος διαπιστώσαμε ότι CSA των 12 bit είναι σχεδόν 3 φορές ταχύτερος από τον 12 bit RCA. Εκτός από την ταχύτητα, ο CSA υλοποιείται εύκολα με ένα μεγαλύτερο κόστος στη κατασκευή του. Πρακτικά, ο CSA είναι μια βελτιωμένη έκδοση του RCA.

Aπό την άλλη μεριά, ο CSA καταλαμβάνει μεγαλύτερη έκταση (συνεπώς μεγαλύτερη ισχύ) λόγω του αριθμού πυλών που έχει. Είναι περισσότερο πολύπλοκος και χρησιμοποιεί και άλλα ψηφιακά στοιχεία όπως ο πολυπλέκτης χωρίς αυτό να σημαίνει κάτι.

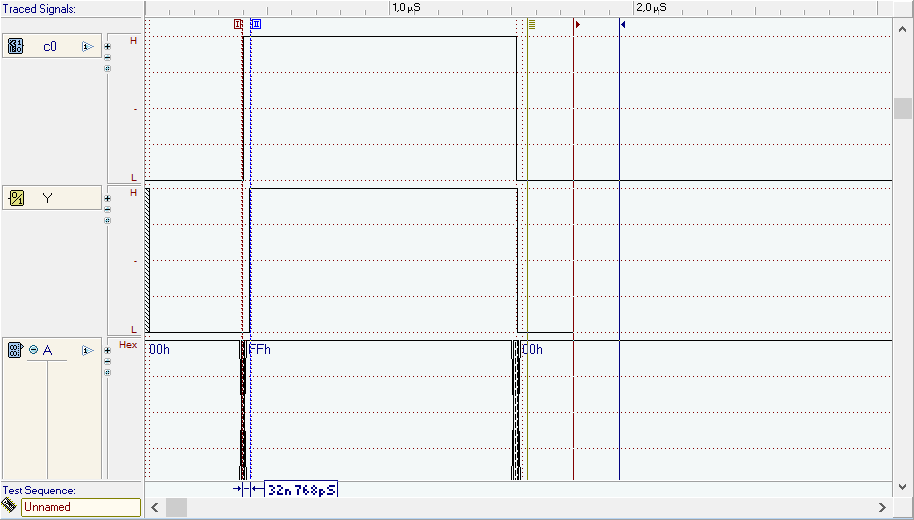
**3.2)Καθυστέρηση (delay) Carry Skip Adder.**

Για να μετρήσουμε την καθυστέρηση του CSA θα αναφερθούμε πάλι στο ίδιο παράδειγμα χρησιμοποιώντας το DEEDS. Έστω ότι έχουμε ένα 8 bit CRS και θέλουμε να υπολογίσουμε το άθροισμα των αριθμών 00000000 και 11111111 με κρατούμενο εισόδου 1 καθώς και να υπολογίσουμε την συνολική καθυστέρηση.

Τα αποτελέσματα που προέκυψαν από το DEEDS είναι τα εξής:

Kαθυστέρηση (Delay):32 ns

Tύπος Αθροιστή: Carry Skip Adder 8 bit



**Σχήμα 3.3: Delay of 8 bit Carry Skip Adder**

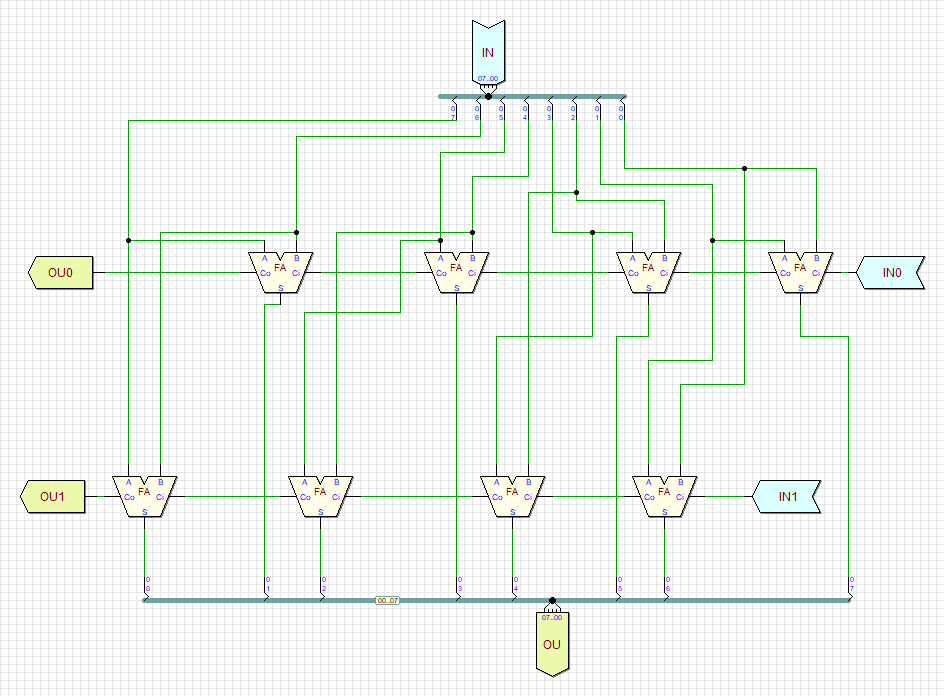
O τύπος που υπολογίζει την καθυστέρηση ενός Carry Skip Adder είναι ο εξής:



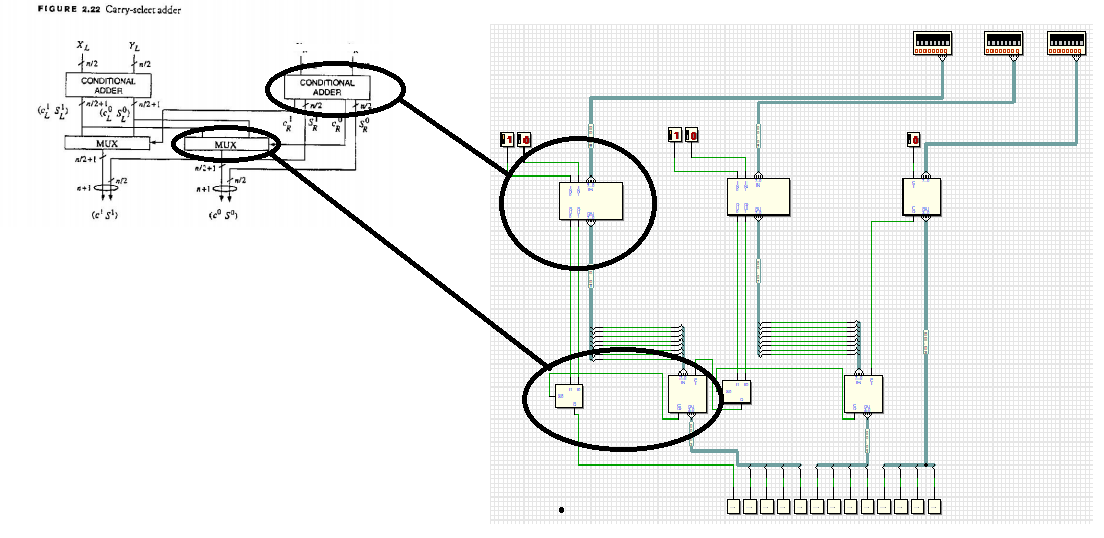
Όπου m είναι ο αριθμός των bits του block και n το πλήθος των bits του αρχικού αριθμού.

**4.0) Carry Select Adder (CSA)**

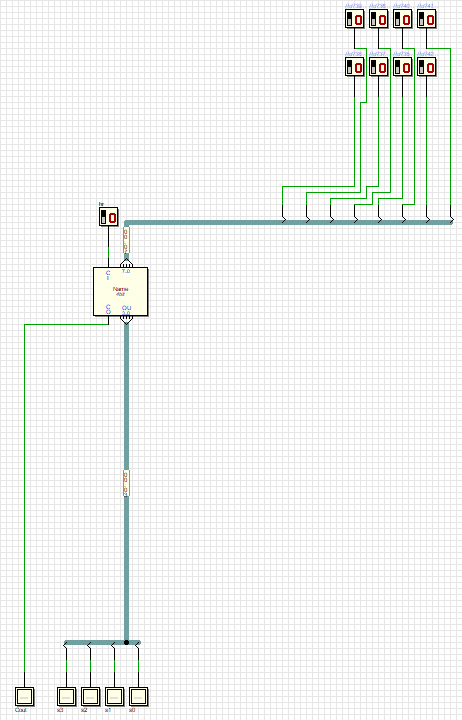
Ο Carry-Select-Adder είναι ένας τύπος αθροιστή, απλός αλλά αρκετά γρήγορος. Ο αθροιστής carry-select-adder αποτελείται γενικά από δύο παράλληλους αθροιστές carry- ripple και έναν πολυπλέκτη. Αναλόγως με τα πόσα bits θέλουμε να έχουμε στην έξοδο μας αυξάνουμε το πλήθος των αθροιστών και των πολυπλεκτών. Για 2ν αθροιστές έχουμε ν πολυπλέκτες. Η προσθήκη δύο αριθμών n-ψηφίων με έναν αθροιστή carry-select γίνεται με δύο προσθήκες (συνεπώς δύο διαδοχικές μετατοπίσεις), για να εκτελεστεί ο υπολογισμός δύο φορές, μία φορά με την παραδοχή ότι το carry-in είναι μηδέν και το άλλο υποθέτοντας θα είναι ένα. Αφού υπολογιστούν τα δύο αποτελέσματα, το σωστό άθροισμα, καθώς και η σωστή εκτέλεση, επιλέγονται στη συνέχεια με τον πολυπλέκτη μόλις γίνει γνωστή η σωστή μεταφορά. Άρα προ-υπολογίζει την είσοδο αν έχουμε 0 ή 1 και παίρνουμε αναλόγως με το (C in) αν είναι 0 ή 1. Επίσης ο αθροιστής χωρίζεται σε Ν κομμάτια όπου Ν=ceil (n/4). ∆ηλαδή αποτελείται από δυάδες αθροιστών 4 ψηφίων (bits).



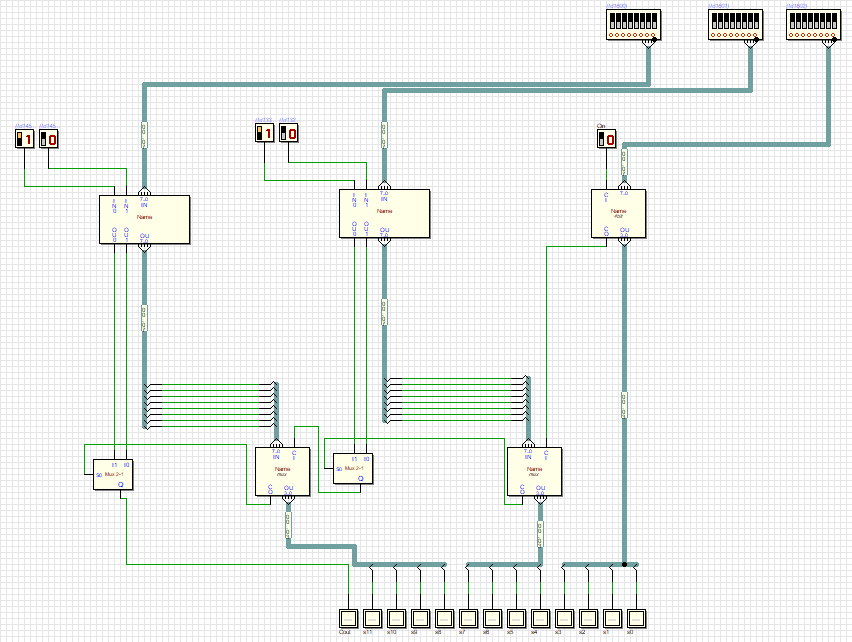
**Σχήμα 4.1: 4 bit CSA**

****

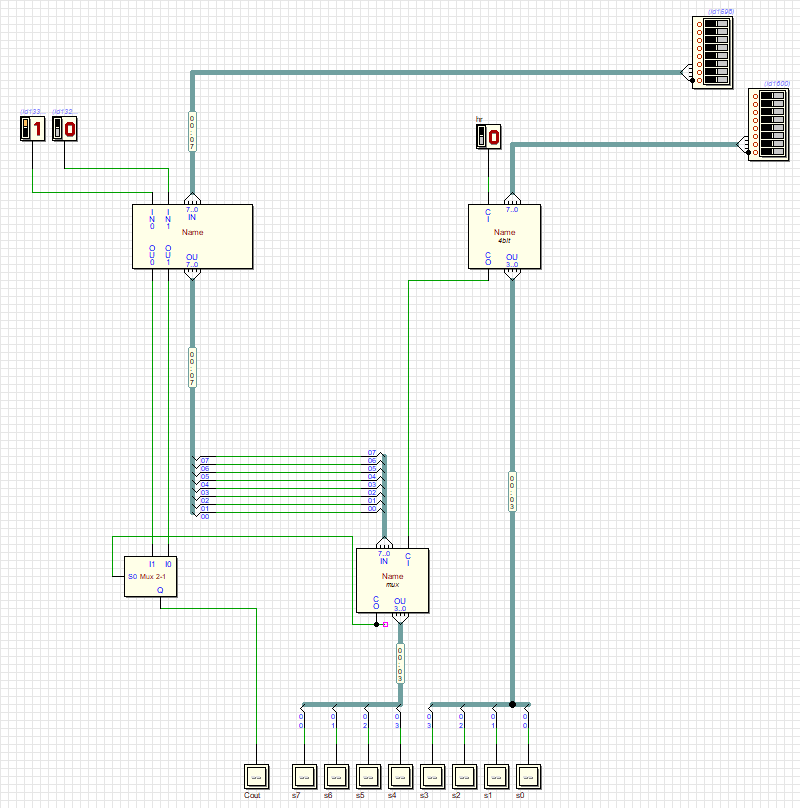
**Σχήμα 4.2: Carry Select Adder**

**Carry- select-adder 4 bits:**Στα 4 bits δε χρειαζόμαστε πολυπλέκτη γιατί ξέρουμε ήδη το C in που βασικά είναι ίδιο με τον Ripper.

**Σχήμα 4.3 CLA 4 bit**



**Σχήμα 4.4: 8 bit Carry Select Adder σχεδιασμένος στο λογισμικό DEEDS**



**Σχήμα 4.5: 12 bit CLA**

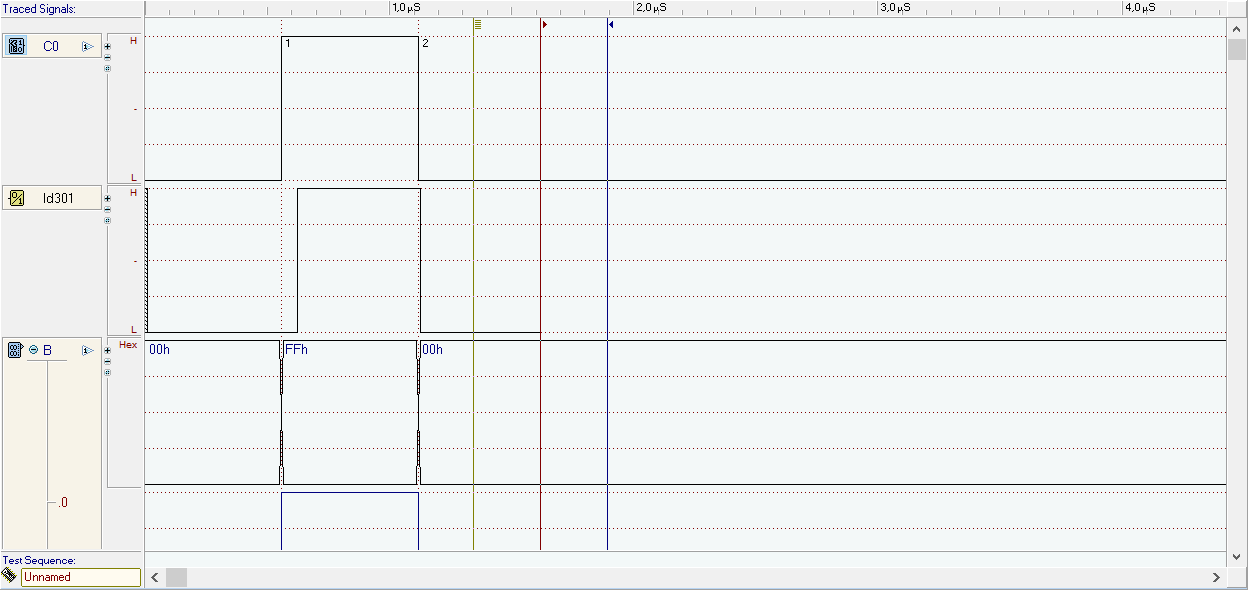
**4.1) Πλεονεκτήματα -Μειονεκτήματα Carry Select Adder**

Ο Carry Select Adder είναι ένας πολύ χρήσιμος αθροιστής λόγω της αξιοπιστίας και της ταχύτητάς του .Υπολογίζει όλα τα δυνατά ενδεχόμενα και αναλόγως με το κρατούμενο εισόδου καθορίζει την έξοδο .Το κρατούμενο εξόδου υπολογίζεται ταχύτατα. Το κύριο μειονέκτημα του είναι η έκταση που καταλαμβάνει (καταναλώνει μεγαλύτερη ισχύ) και το αυξημένο hardware που διαθέτει. Η υλοποίησή του απαιτεί αρκετές πύλες και πολυπλέκτες και συνεπώς αυξημένο κόστος.

**4.2) Καθυστερήση Carry Select Adder**

Για να μετρήσουμε την καθυστέρηση του CSA θα αναφερθούμε πάλι στο ίδιο παράδειγμα χρησιμοποιώντας το DEEDS. Έστω ότι έχουμε ένα 8 bit CRS και θέλουμε να υπολογίσουμε το άθροισμα των αριθμών 00000000 και 11111111 με κρατούμενο εισόδου 1 καθώς και να υπολογίσουμε την συνολική καθυστέρηση.

Τα αποτελέσματα που προέκυψαν από το DEEDS είναι τα εξής:



**Σχήμα 4.6: Καθυστέρηση σε 8 bit CSA**

Όπως και βλέπουμε στο παραπάνω σχήμα η κυματομορφή εξόδου που συμβολίζεται με Ιd301 εμφανίζει χρονική καθυστέρηση με την είσοδο. Οι καθυστερήσεις και των τεσσάρων αθροιστών θα παρουσιαστούν αναλυτικά στο τέλος της αναφοράς σε ιστόγραμμμα.

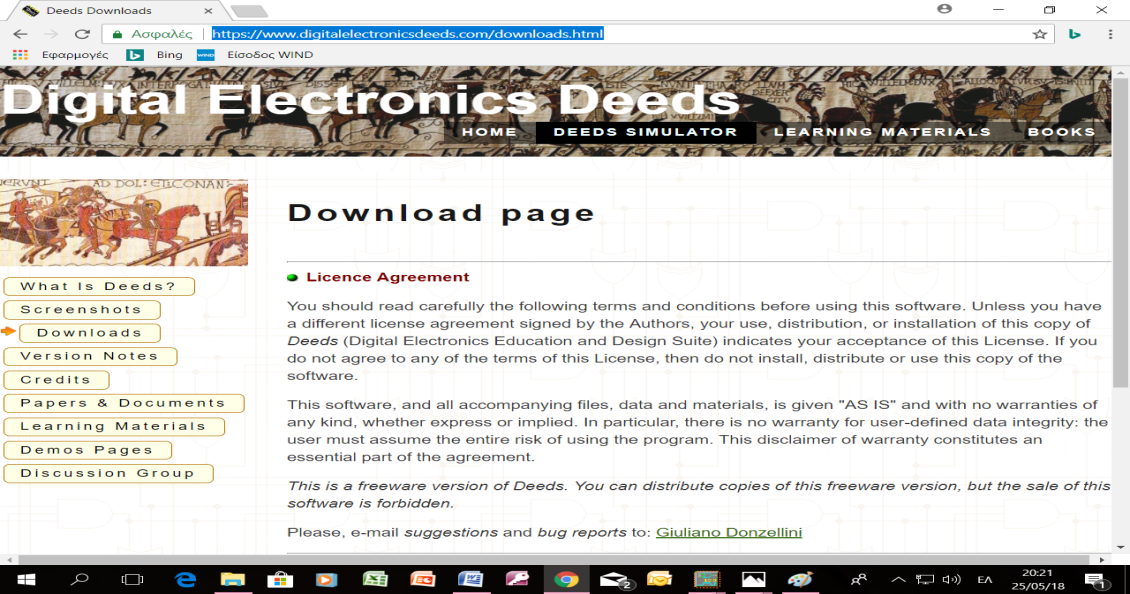
Θεωρητικά,η συνολική καθυστέρηση του CSA υπολογίζεται από τον παρακάτω τύπo:

**5.0)Digital Electronics Deeds**

Το DEEDS είναι ένα ολοκληρωμένο πρόγραμμα που διατίθεται δωρεάν σε όποιον επιθυμεί να το εγκαταστήσει. Στόχος, αυτού του προγράμματος είναι η κατανόηση και η προσομοίωση ψηφιακών κυκλωμάτων που θα βοηθήσει τον χρήστη να εμβαθύνει και να διευρύνει τις γνώσεις του στα ψηφιακά κυκλώματα. Eίναι αρκετά απλό στη χρήση του και παρέχει μεγάλες δυνατότητες στο χρήστη όπως σχεδίαση και προσομοίωση κυκλωμάτων, απεικόνιση γραφημάτων, υπολογισμός καθυστερήσεων, δημιουργία δικών μας ψηφιακών στοιχείων, αποθήκευση και εκτύπωση κυκλωμάτων.

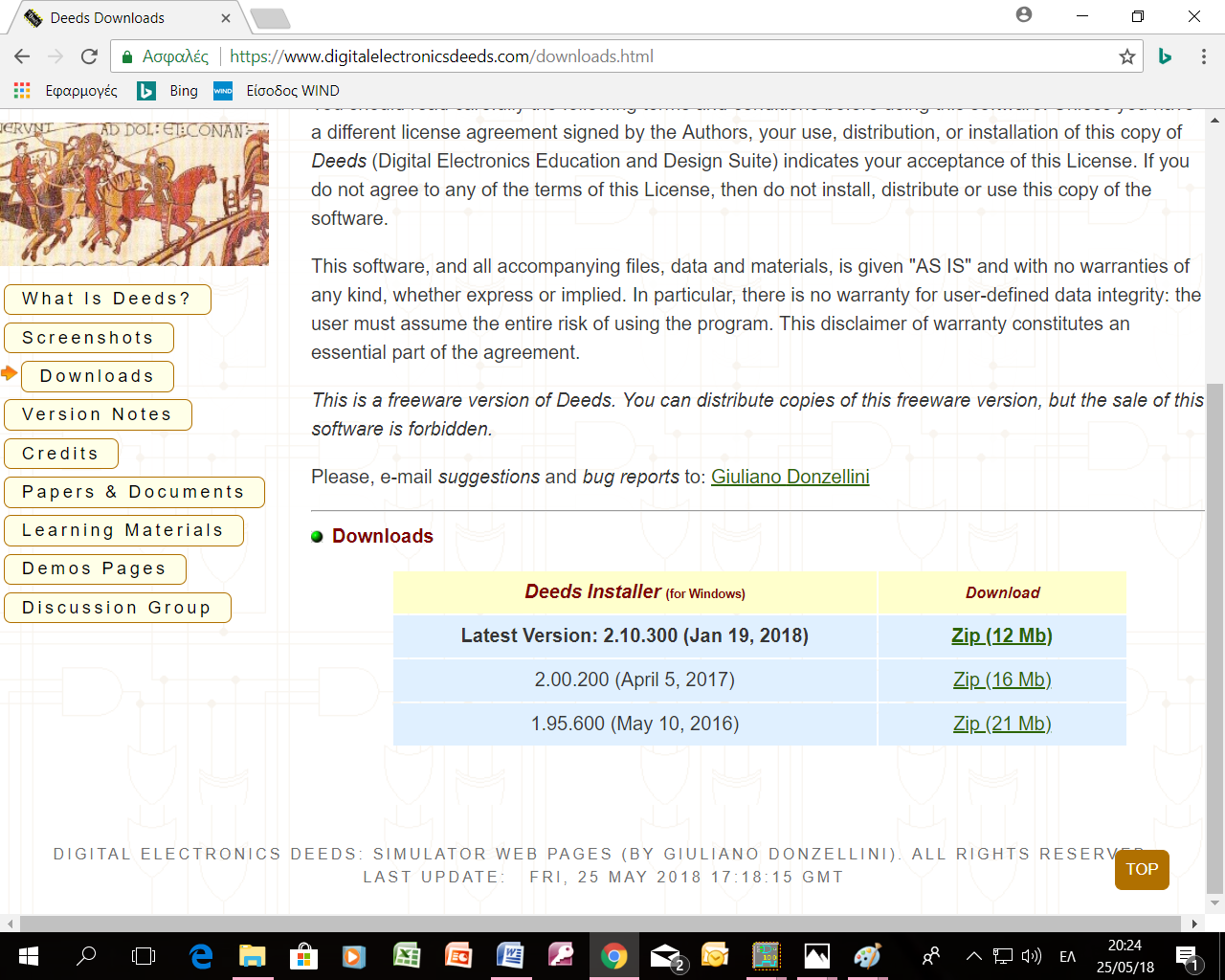
Παρακάτω αναφέρονται τα βήματα που πρέπει να ακολουθήσουμε για να εγκαταστήσουμε το DEEDS και να εισέλθουμε στην βασική πλατφόρμα υλοποίησης ψηφιακών κυκλωμάτων.

1)Πληκτρολογούμε το link <https://www.digitalelectronicsdeeds.com/downloads.html> στο google και το πατάμε.



**Σχήμα 5.1: Επίσημη Ιστοσελίδα του Digital Electronics Deeds**

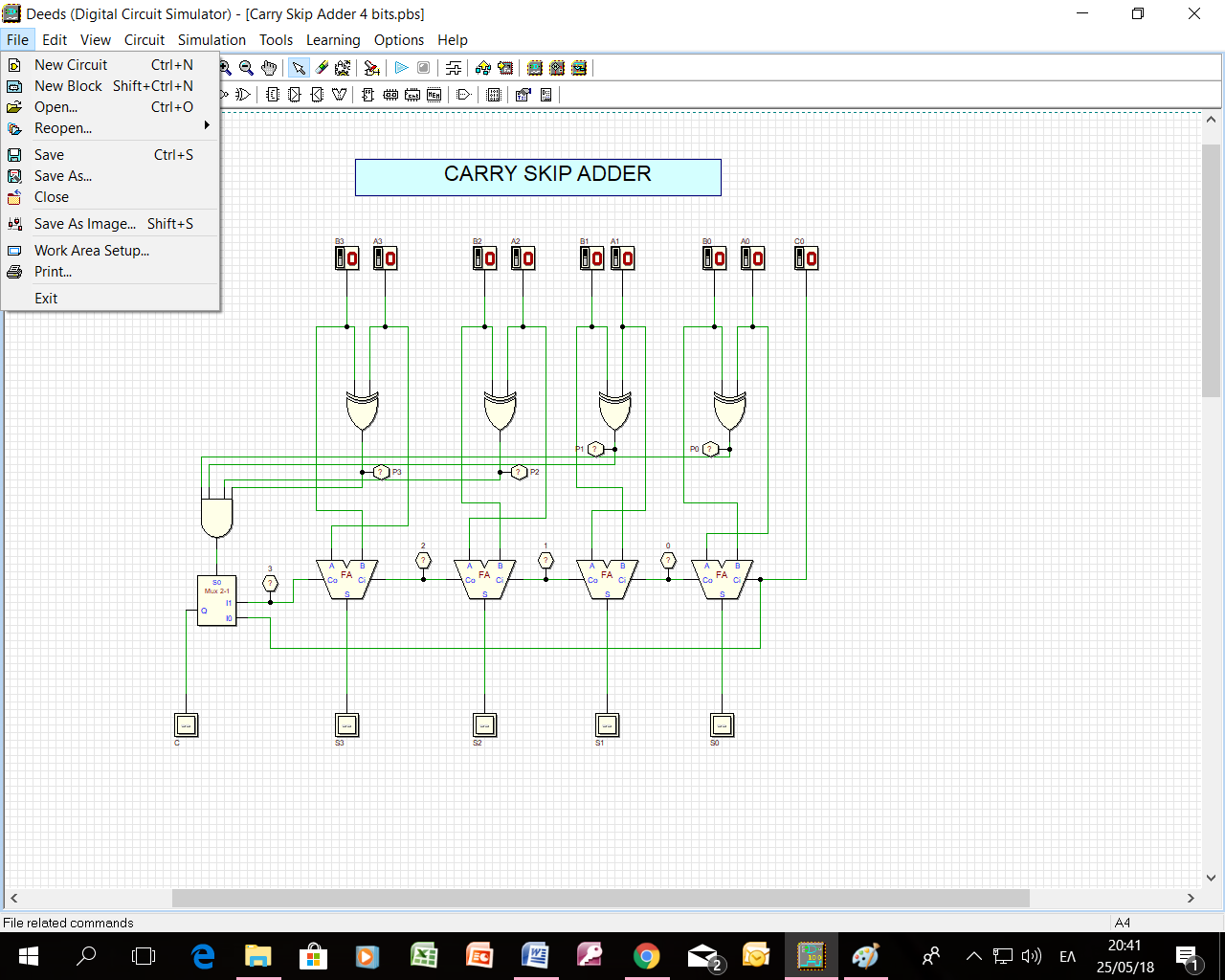
2)Διαλέγουμε στο κάτω μέρος της οθόνης την επιλογή Latest Versio:2.10.300(Jan 19,2018) ΖΙP.(12 MB)



**Σχήμα 5.2: Διαθέσιμες εκδόσεις για Windows του Deeds**

3)H εγκατάσταση του προγράμματος ολοκληρώθηκε .Στην επιφάνεια εργασίας του Η\Υ μας δημιουργήθηκαν δύο εικονίδια. Πατάμε το εικονίδιο Deeds-DcS και εμφανίζεται ο χώρος που γίνεται η υλοποίηση των κυκλωμάτων.

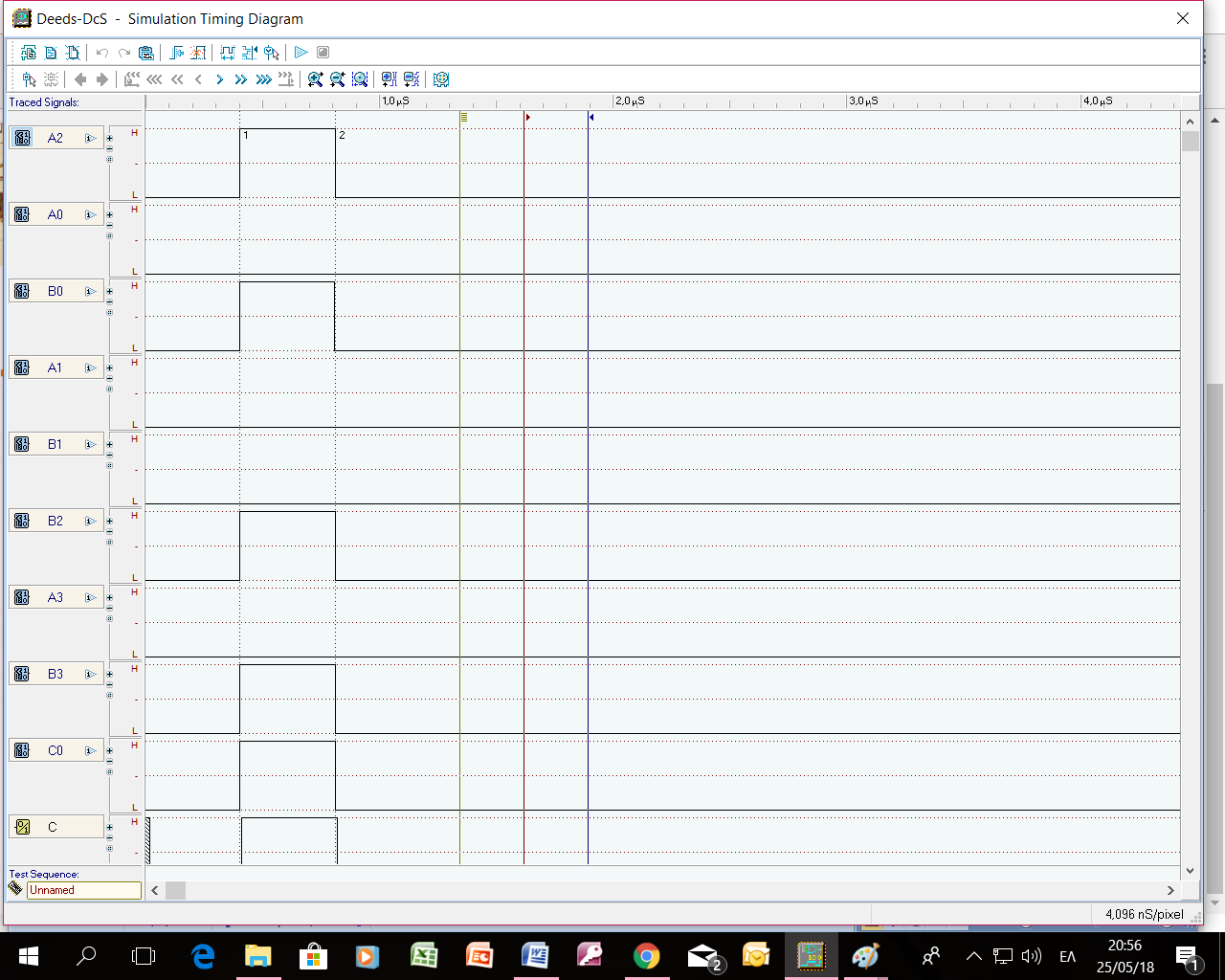
4)Πλέον μπορούμε να δημιουργήσουμε το δικό μας κύκλωμα. Στο επάνω μέρος βρίσκονται όλες οι πύλες που θέλουμε να χρησιμοποιήσουμε ,έτοιμα ψηφιακά κυκλώματα όπως πολυπλέκτες ,κωδικοποιητές,flip-flop και πολλά άλλα. Επίσης , από το πάνω μέρος μπορούμε να χρησιμοποιήσουμε εισόδους, εξόδους, ζεύξεις και πολλά διάφορα άλλα στοιχεία.



**Σχήμα 5.3:Ένα παράδειγμα που αφορά στην υλοποίηση ενός απλού κυκλώματος**

5)Για να αποθηκεύσουμε ένα κύκλωμα επιλέγουμε File->Save και επιλέγουμε σε ποιο μέρος θα το αποθηκεύσουμε και με τι όνομα. Αν θέλουμε να ανοίξουμε ένα αρχείο πατάμε το εικονίδιο Deeds-DcS και στη συνέχεια τις επιλογές File->Open και ανοίγουμε το αρχείο που έχουμε δημιουργήσει.

Εάν θέλουμε να τρέξουμε το πρόγραμμα πατάμε το μπλε τρίγωνο που βρίσκεται πάνω δεξιά. Δίπλα από το μπλε τρίγωνο υπάρχει ένας κόκκινος φακός που η λειτουργία του είναι να ελέγχει σφάλματα του κυκλώματος. Τέλος, επιλέγοντας την επιλογή που βρίσκεται δύο θέσεις δεξιά του μπλε τριγώνου δίνεται η δυνατότητα για μετρήσεις. Για παράδειγμα, στο σχήμα 5.3 απεικονίζεται ένας 4 bit Carry Skip Adder στο οποίο μπορούμε να μετρήσουμε την καθυστέρηση διάδοσης του κρατουμένου.



**Σχήμα 5.4 :Eισαγωγή τετραγωνικών παλμών**

Η μέτρηση της καθυστέρησης θα πραγματοποιηθεί με την επιλογή δύο κάθετων κερσορων .Ο πρώτος κέρσορας θα δείχνει στη λήξη των σημάτων εισόδου που εισάγονται ταυτόχρονα ,ενώ ο δεύτερος στο παλμό εξόδου που λογικά θα έχει μετατοπιστεί προς τα δεξιά. Αυτή η μετατόπιση ισούται με την καθυστέρηση.

**6.0)ΒΙΒΛΙΟΓΡΑΦΙΑ**

**[1]** Ψηφιακή Σχεδίαση Με Εισαγωγή Στη Verilog HDL M.MORRIS MANO | MICHAEL D.CILETI

**[2]** Ψηφιακή Σχεδίαση ΑΡΧΕΣ ΚΑΙ ΠΡΑΚΤΙΚΕΣ JOHN F.WAKERLY

**[3]** Digital Arithmetic MILOS D.ERCEGOVAC | TOMAS LANG

**[4] <https://www.digitalelectronicsdeeds.com/>**

**[5]** **<https://en.wikipedia.org/wiki/Carry-skip_adder>**

**[6]** <https://en.wikipedia.org/wiki/Adder_(electronics)>

**[7]** <https://en.wikipedia.org/wiki/Carry-lookahead_adder>