

ΕΡΓΑΣΤΗΡΙΟ ΣΧΕΔΙΑΣΜΟΥ ΟΛΟΚΛΗΡΩΜΕΝΩΝ
ΚΥΚΛΩΜΑΤΩΝ - VLSI II

WALLACE MULTIPLIER 8-bit

ΟΜΑΔΑ 93

Αρχοντά Χριστίνα 7219

Παπαϊωάννου Μαγδαληνή 7359

ΣΚΟΠΟΣ ΤΗΣ ΕΡΓΑΣΙΑΣ

Σκοπός της εργασίας ήταν να συνδυάσουμε τις γνώσεις όλου του εξαμήνου ώστε να σχεδιάσουμε με τη χρήση δύο εργαλείων, του PSPICE και του MODELSIM, έναν πολλαπλασιαστή Wallace των 8-bit.

WALLACE MULTIPLIER

Οι πολλαπλασιαστές είναι σημαντικό κομμάτι των συστημάτων επεξεργασίας ψηφιακών σημάτων, και ως εκ τούτου σημαντικές πηγές καθυστέρησης.

Αναζητούνται λοιπόν μέθοδοι επιτάχυνσης των πολλαπλασιαστών. Μια αποδοτική μέθοδος είναι η πρόσθεση των μερικών γινομένων του πολλαπλασιασμού σε στήλες.

Αυτό μπορεί να γίνει με χρήση του Wallace tree, το οποίο απαιτεί $\log_3\left(\frac{N}{2}\right)$

επίπεδα αθροιστών για τη μείωση των N εισόδων σε 2 εξόδους πλεονασματικής μορφής αθροίσματος-κρατούμενου. Το Wallace tree προσθέσει δηλαδή τα μερικά γινόμενα παράλληλα, αντί διαδοχικά, μειώνοντας έτσι το critical path και τον συνολικό αριθμό των adders.

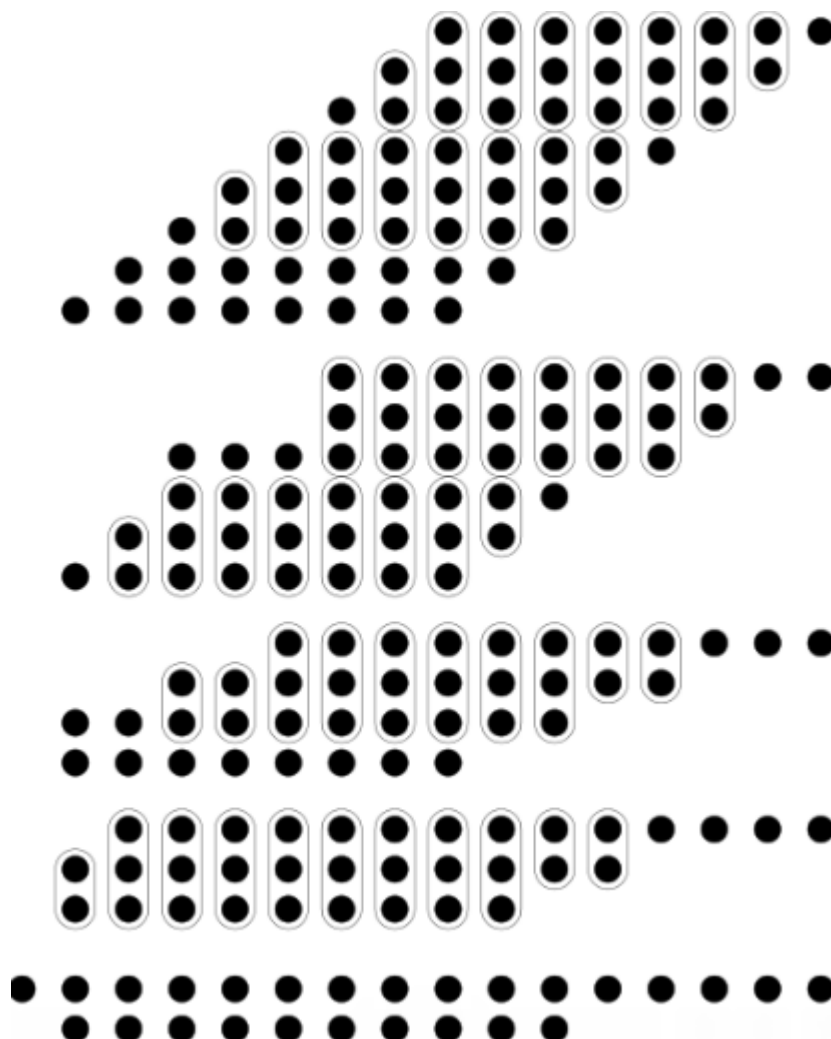
Το Wallace tree αποτελείται από τρία βήματα:

- Πράξη AND μεταξύ όλων των bit κάθε αριθμού με αποτέλεσμα να πάρουμε $8*8=64$ μερικά γινόμενα των οποίων τα βάρη εξαρτώνται από τη θέση τους.
- Μείωση του αριθμού των γραμμών μερικών γινομένων (PP) σε 2 με την χρήση full και half adders.
- Πρόσθεση των bit που έχουν ίδιο βάρος με έναν Carry Select Adder.

Για να μεταβούμε σε 2 μόνο γραμμές μερικών γινομένων πρέπει όποτε υπάρχουν 3 ή περισσότερα μερικά γινόμενα με το ίδιο βάρος να τα προσθέτουμε με έναν full adder και το sum που προκύπτει θα έχει το ίδιο βάρος ενώ το carry κατά ένα μεγαλύτερο. Σε περίπτωση που υπάρχουν 2 μερικά γινόμενα με το ίδιο βάρος και δεν πρόκειται να αυξηθούν σε επόμενο στάδιο, τα προσθέτουμε με half adder και ισχύει το ίδιο για τα βάρη των εξόδων. Τέλος, εάν ένα μόνο μερικό γινόμενο, το περνάω αυτούσιο στο επόμενο layer.

Για την τελική πρόσθεση χρησιμοποιούμε έναν Carry Select Adder με στόχο την ταχύτητα, εις βάρος του υλικού και της επιφάνειας, καθώς ο CSA κάνει προσθέσεις των 4bit για κάθε πιθανό Cin και στη συνέχεια διαλέγει το σωστό αποτέλεσμα μέσω πολυπλεκτών.

Ένα παράδειγμα πολλαπλασιασμού δύο 8-bit μη-προσημασμένων αριθμών είναι αυτό:



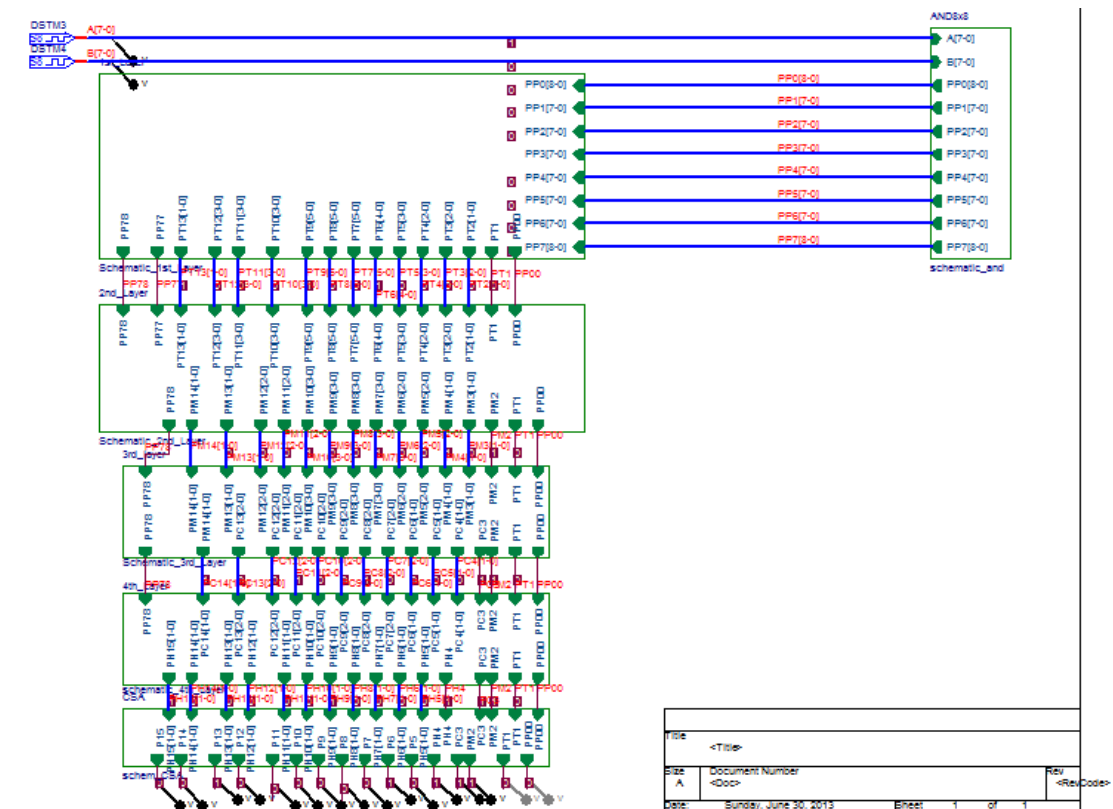
Στην περίπτωση μας οι αριθμοί είναι προσημασμένοι, σε αναπαράσταση συμπληρώματος ως προς 2.

Είναι δυνατό, αντί να προσθέσουμε το πρόσημο στα αριστερά κάθε γραμμής των μερικών γινομένων (PP), να αντιστρέψουμε και να μεταθέσουμε κάθε πρώτο και τελευταίο στοιχείο κάθε στήλης, προσθέτοντας στην πρώτη και τελευταία γραμμή έναν άσσο επί πλέον στα αριστερά (αλγόριθμος Baugh-Wooley). Μειώνεται μ' αυτό τον τρόπο αισθητά ο αριθμός των απαιτούμενων ψηφίων των μερικών γινομένων.

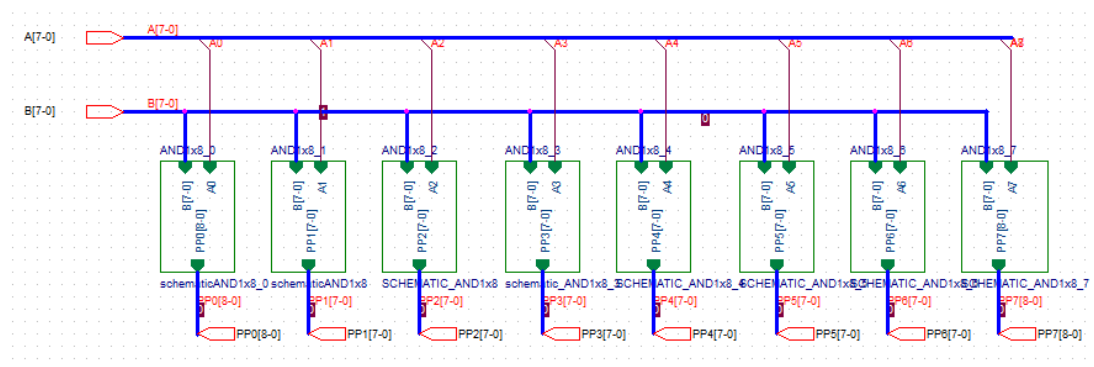
1. PSPICE

Βασιζόμενοι στα παραπάνω φτιάξαμε τον Wallace multiplier στο P-Spice.

Ο σκελετός του πολλαπλασιστή έχει ως εξής:

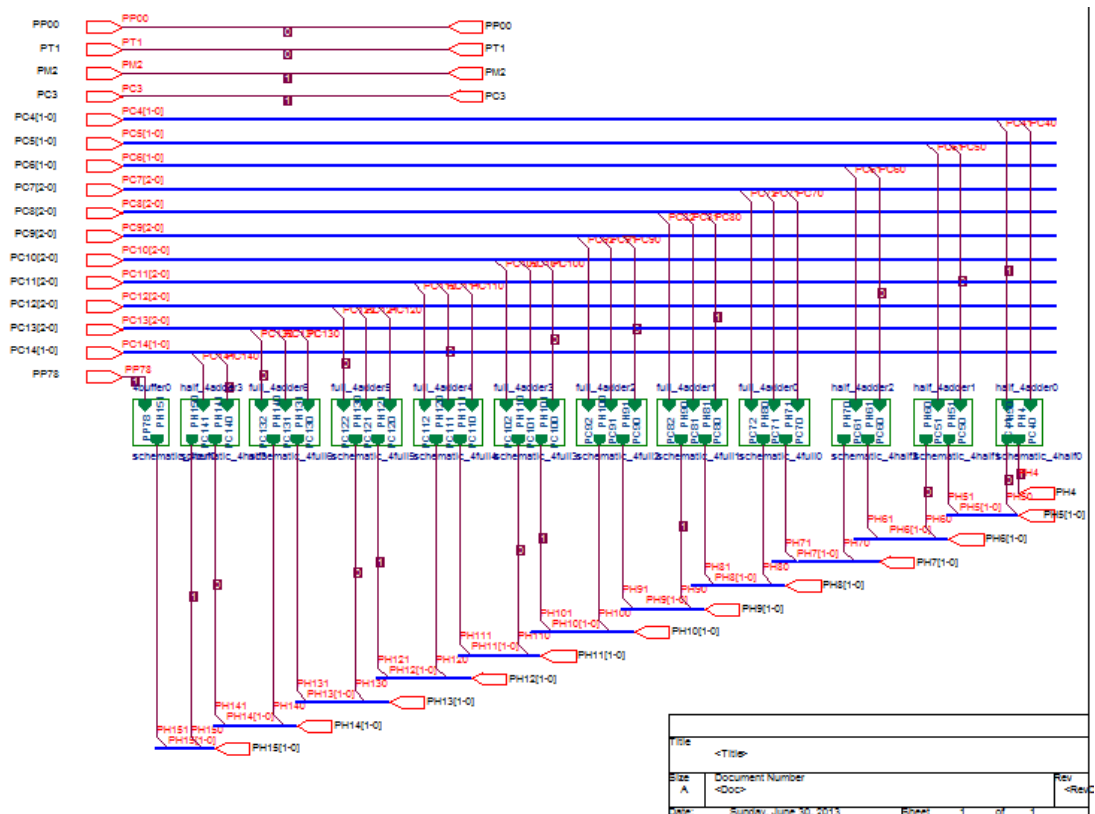


Το πρώτο μπλοκ, πάνω δεξιά, αναπαριστά την πύλη AND8x8, απ' την οποία δημιουργούνται τα μερικά γινόμενα του πολλαπλασιασμού μας:



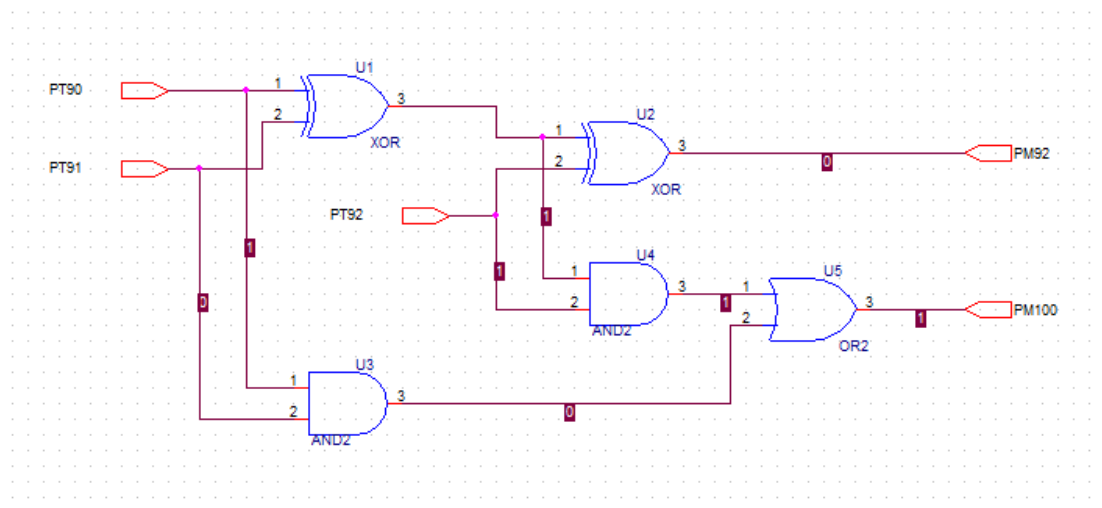
Η οποία αποτελείται από 8 πύλες AND8x1:

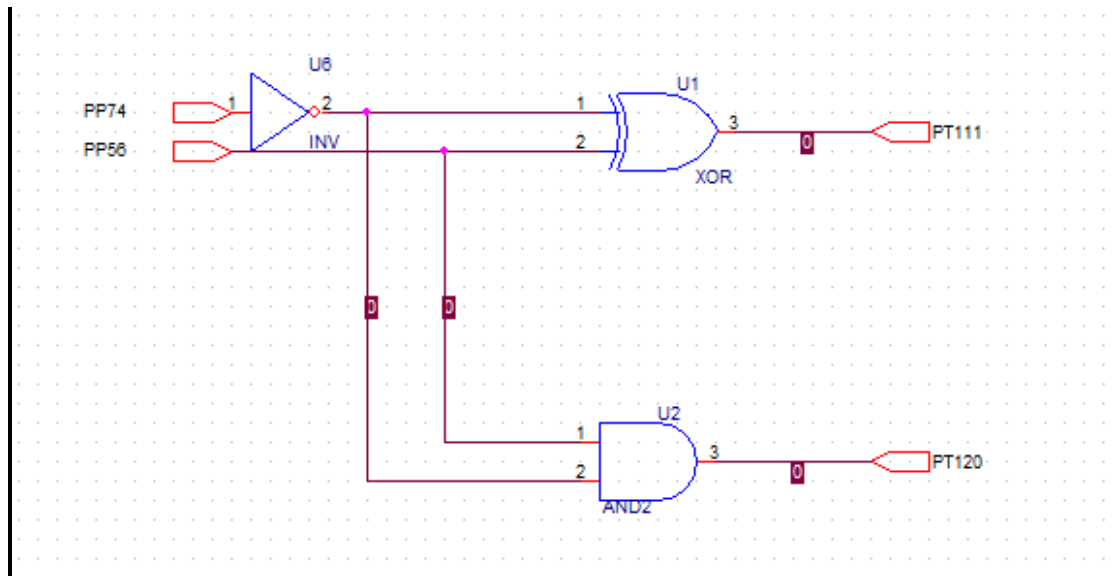
Και το τέταρτο:



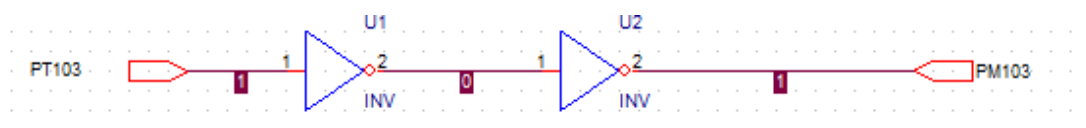
Είναι προφανές πως η διαδικασία απλοποιείται σε κάθε επίπεδο.

Τα παραπάνω επίπεδα αποτελούνται από full adders και half adders:

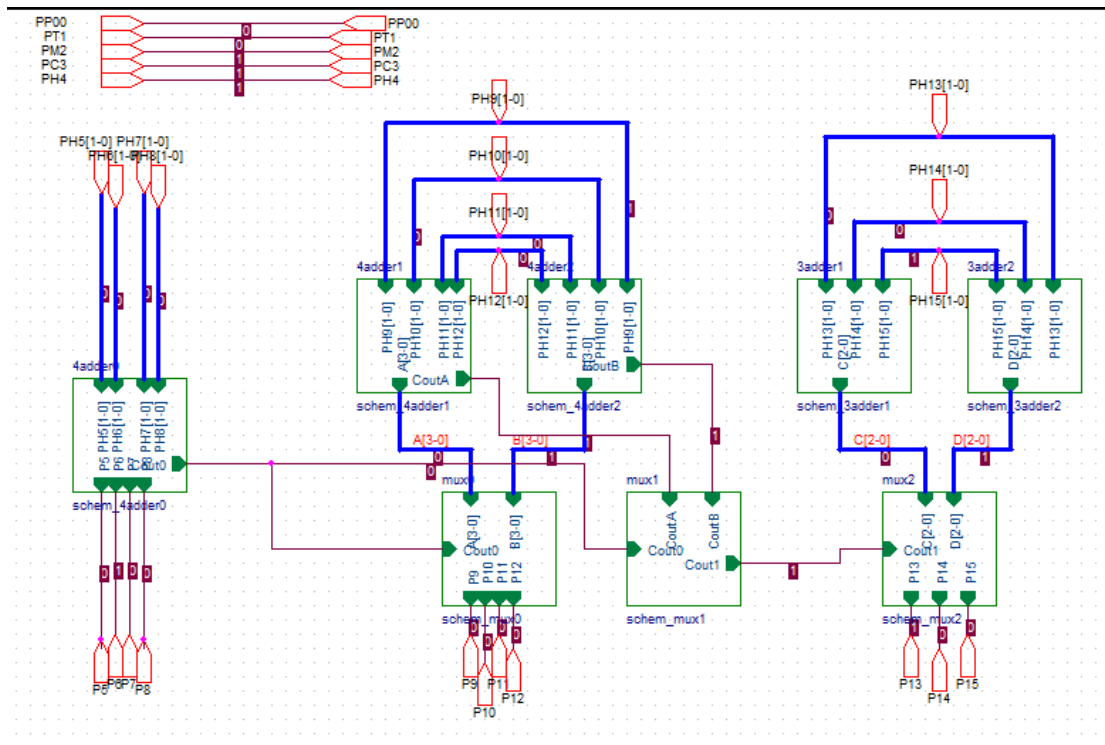




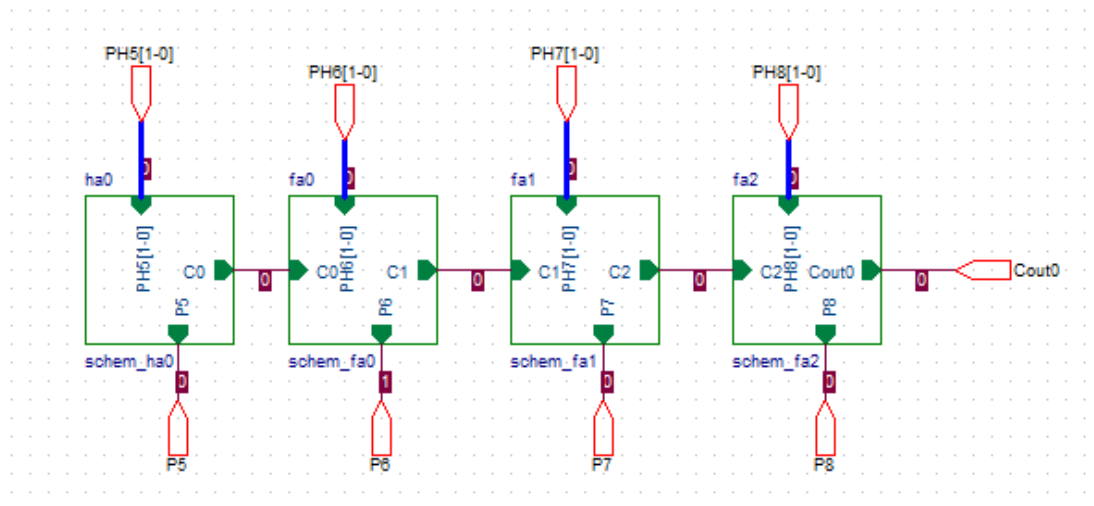
Καθώς και buffers, οι οποίοι βοήθησαν στο να αποφευχθούν τα προβλήματα που δημιουργεί το P-Spice όταν πρέπει μια σύνδεση που ξεκινά από ένα bus να καταλήξει αυτούσια σε κάποιο άλλο bus με το ίδιο όνομα.

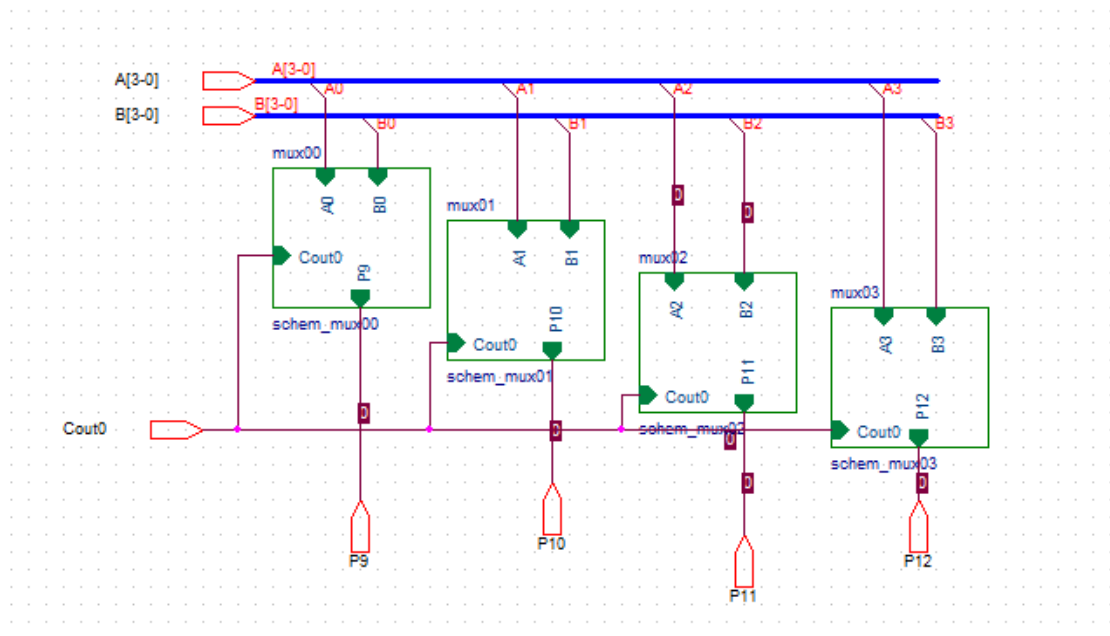


Τελευταίο μπλοκ του πολλαπλασιαστή είναι ένας Carry Select Adder, ο οποίος προσθέτει αποδοτικά τους δύο όρους που προέκυψαν από το τελευταίο επίπεδο αθροιστών και μας δίνει το τελικό αποτέλεσμα.

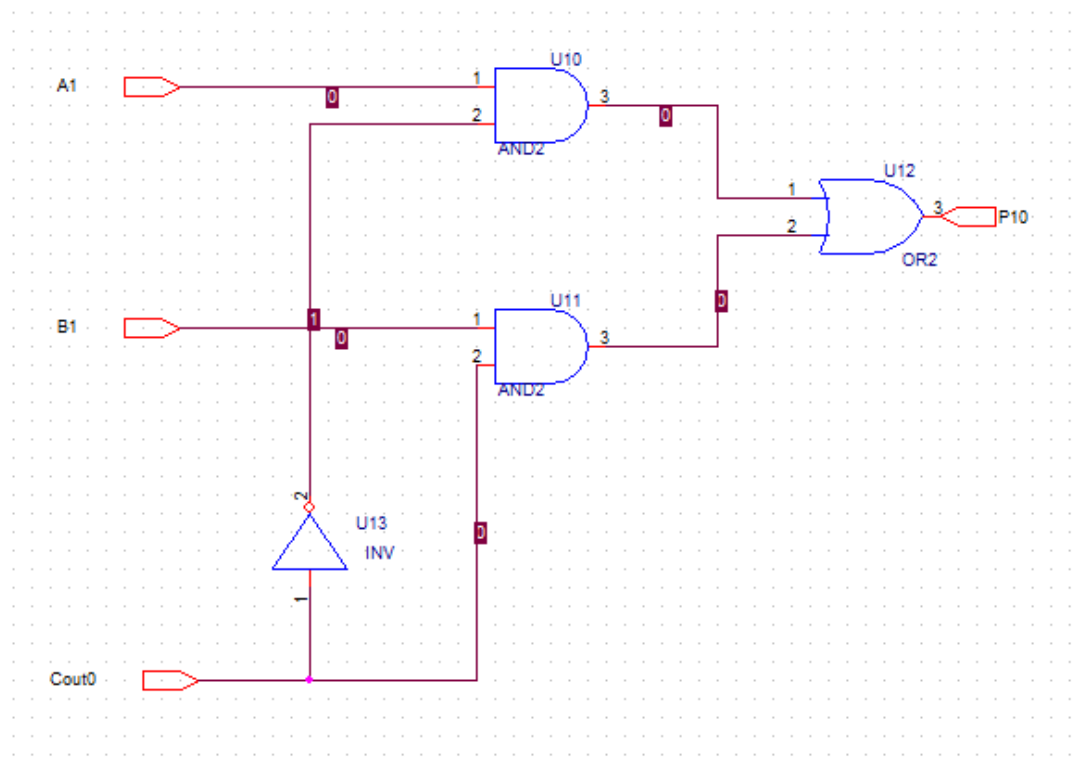


Έχει διααιρεθεί σε adders και πολυπλέκτες των 8bit:



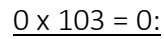


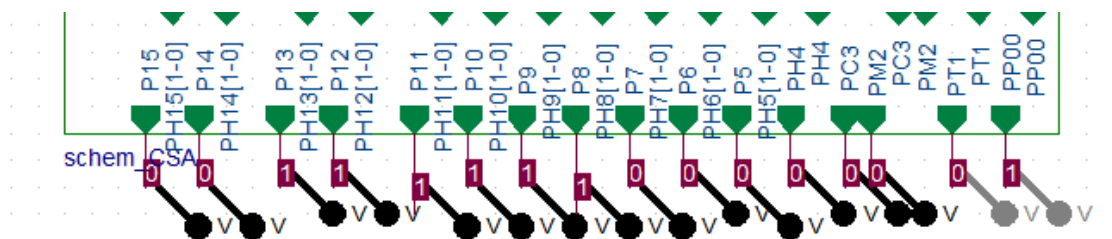
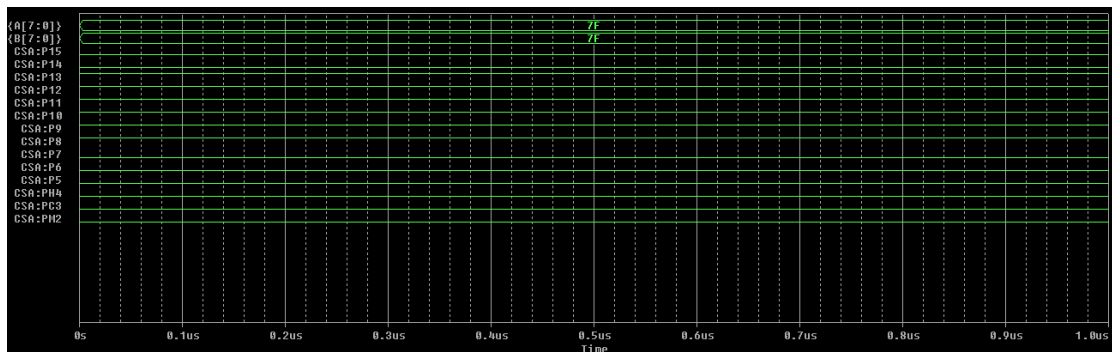
Οι full και half adders είναι υλοποιημένοι όπως παραπάνω ενώ οι πολυπλέκτες των 2x1bit:



2.MODELSIM

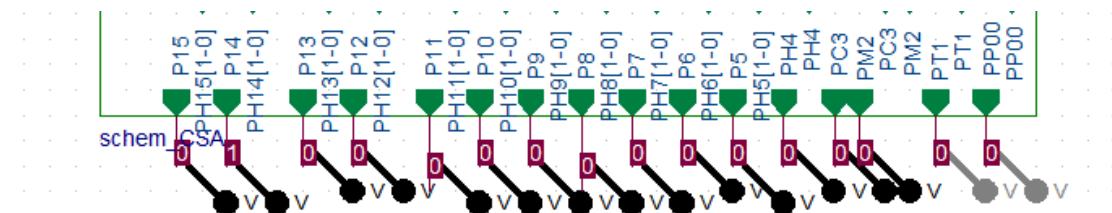
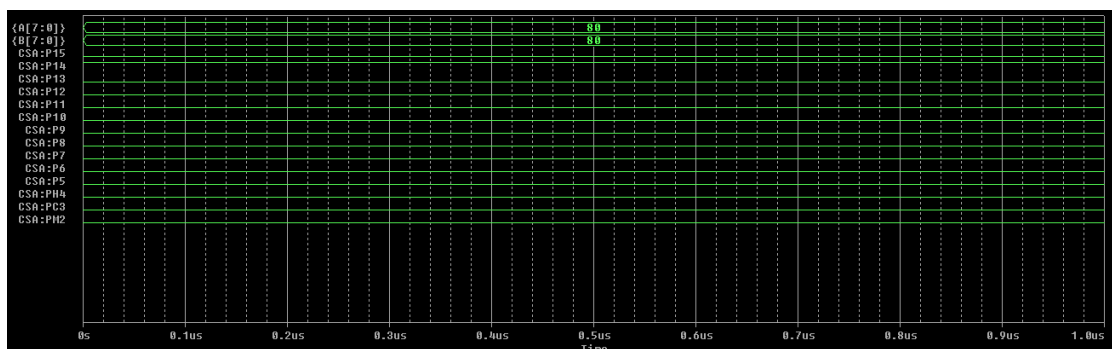
Στο modelsim ο πολλαπλασιαστής υλοποιήθηκε με την ίδια ακριβώς λογική, αλλά αυτή τη φορά με κωδικοποίηση VHDL.





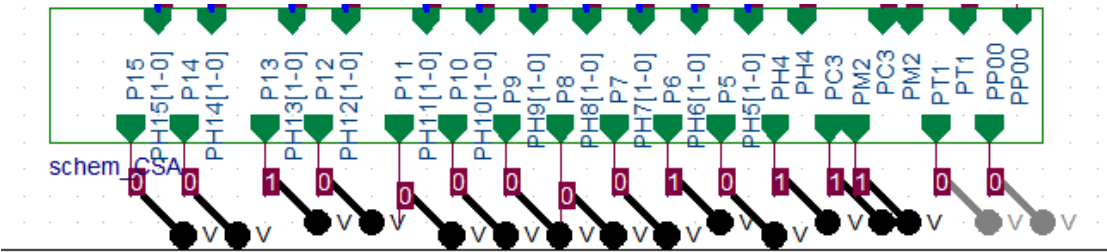
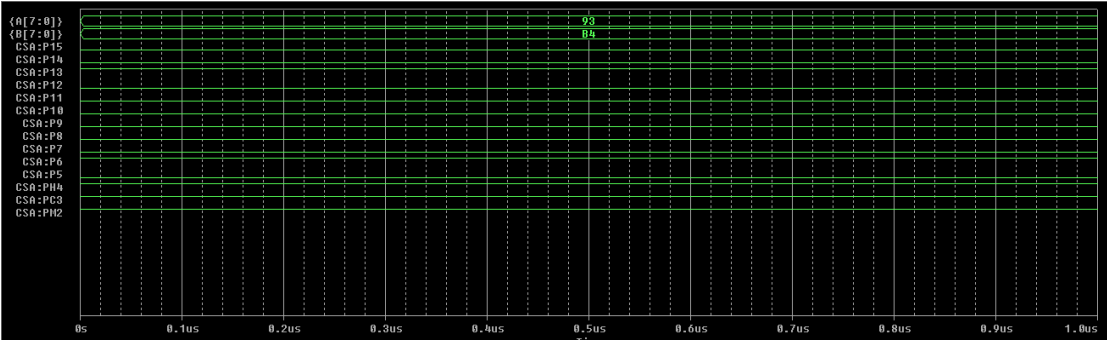
$$-128 \times -128 = 16384 :$$

/wallace_mult/a	10000000	10010011	00000000	11111111	01111111	01110101	10000000
/wallace_mult/b	10000000	01100001	01100111	11111111	01111111	00011010	10000000
/wallace_mult/result	0100000000000000	11010110...	00000000...	00000000...	00111111...	00001011...	01000000...



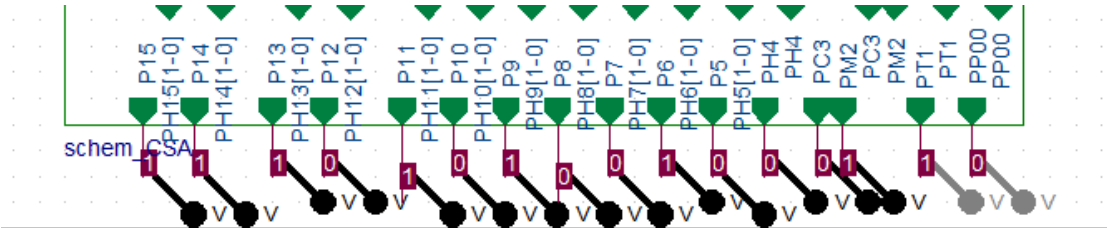
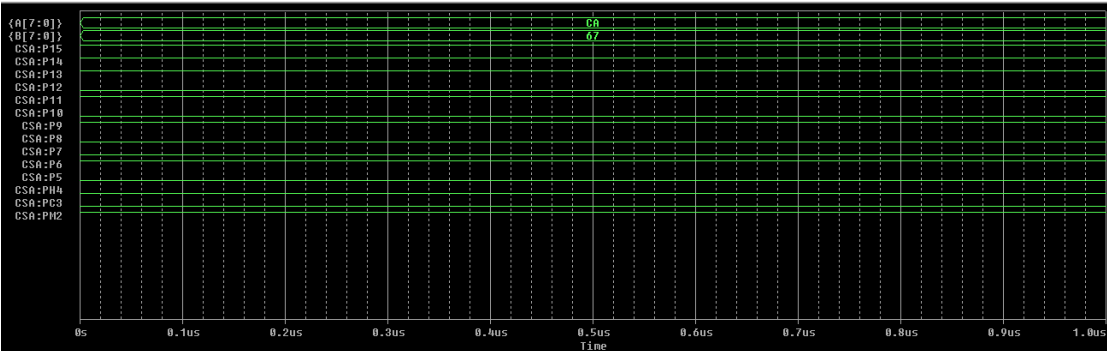
$$-109 \times -76 = 8284 :$$

/wallace_mult/a	10010011	10010011	00000000	11111111	01111111	01110101	10000000	10010011
/wallace_mult/b	10110100	01100001	01100111	11111111	01111111	00011010	10000000	10110100
/wallace_mult/result	0010000001011100	11010110...	00000000...	00000000...	00111111...	00001011...	01000000...	00100000...



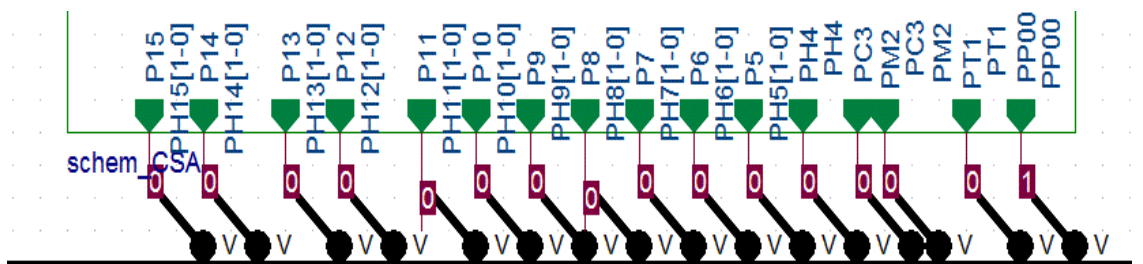
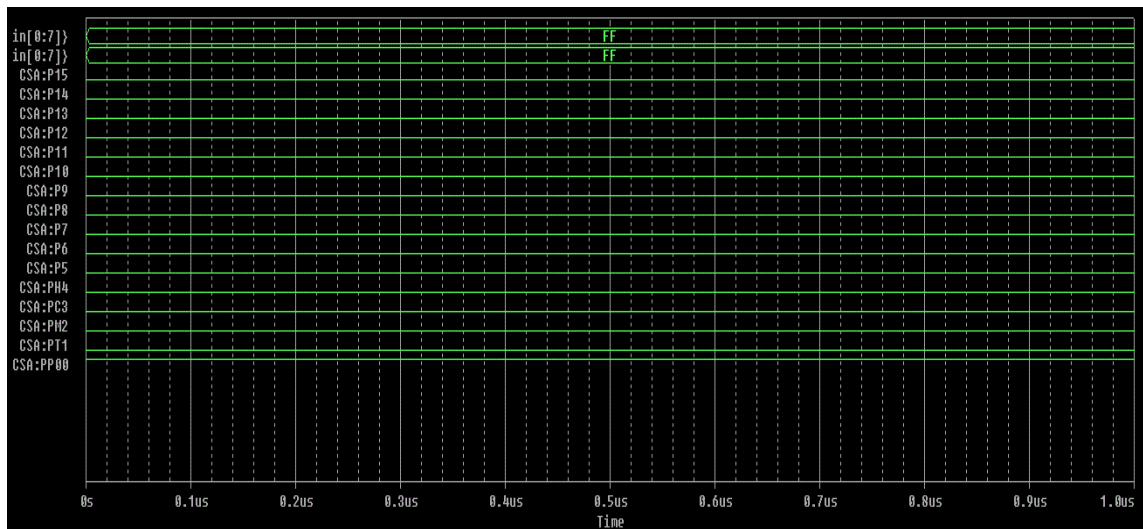
-109 x 97 = -10573:

+ /wallace_mult/a	11001010	10010011	00000000	11111111	01111111	01110101	10000000	10010011	11001010
+ /wallace_mult/b	01100111	01100001	01100111	11111111	01111111	00011010	10000000	10110100	01100111
+ /wallace_mult/result	1110101001000110	11010110...	00000000...	00000000...	00111111...	00001011...	01000000...	00100000...	11101010...



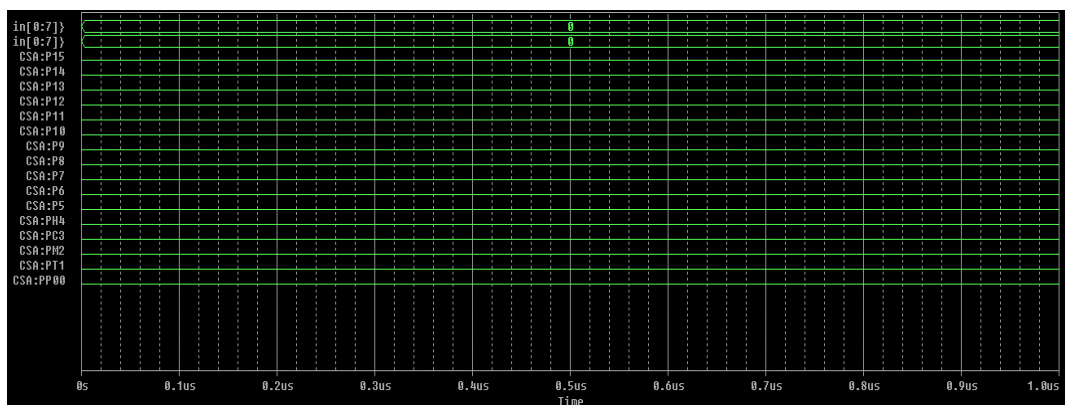
-1 x -1 = 1:

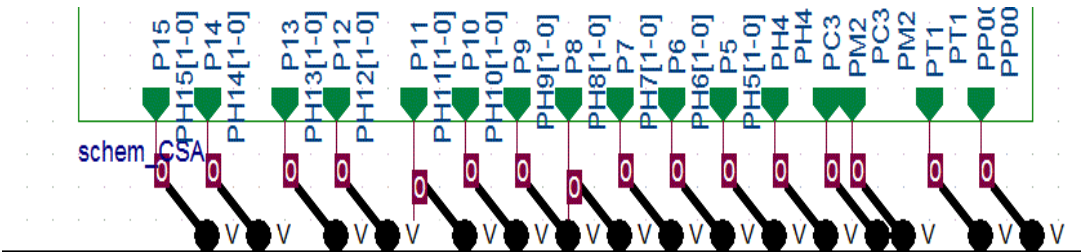
+ /wallace_mult/a	11111111	10010011	00000000	11111111
+ /wallace_mult/b	11111111	01100001	01100111	11111111
+ /wallace_mult/result	0000000000000001	11010110...	00000000...	00000000...



0 x 0 = 0:

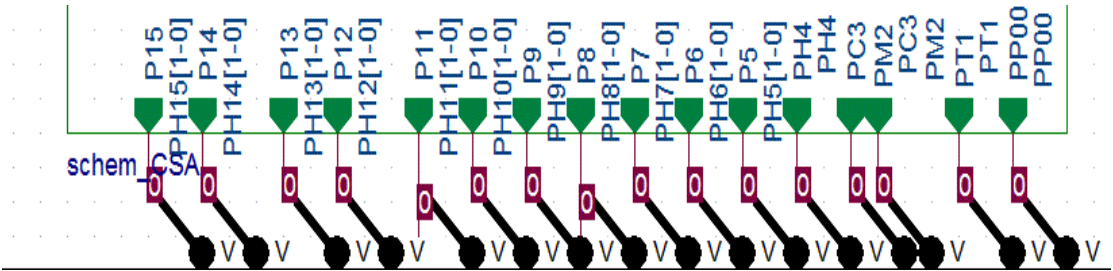
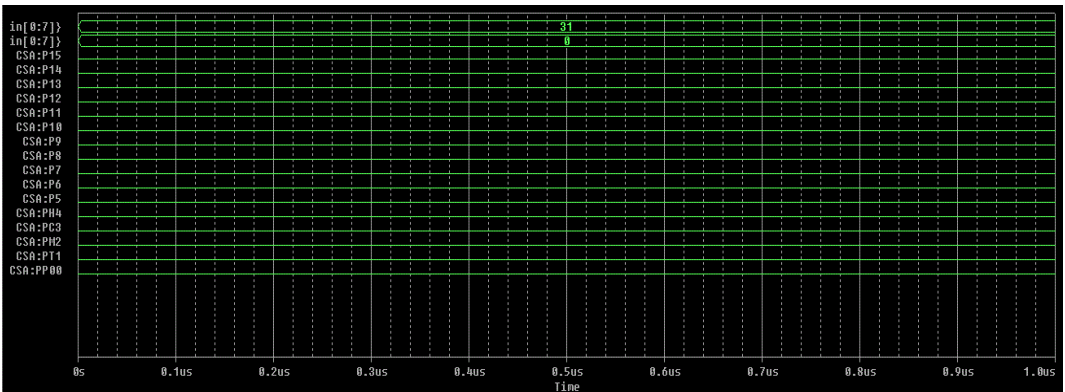
Messages		
+ /wallace_mult/a	00000000	00000000
+ /wallace_mult/b	00000000	00000000
+ /wallace_mult/result	0000000000000000	0000000000000000





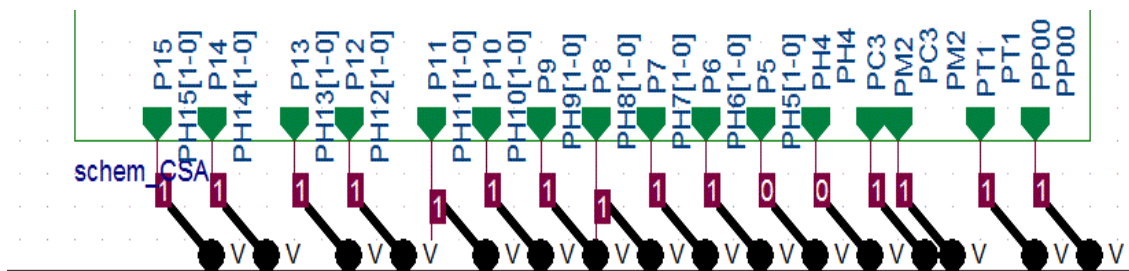
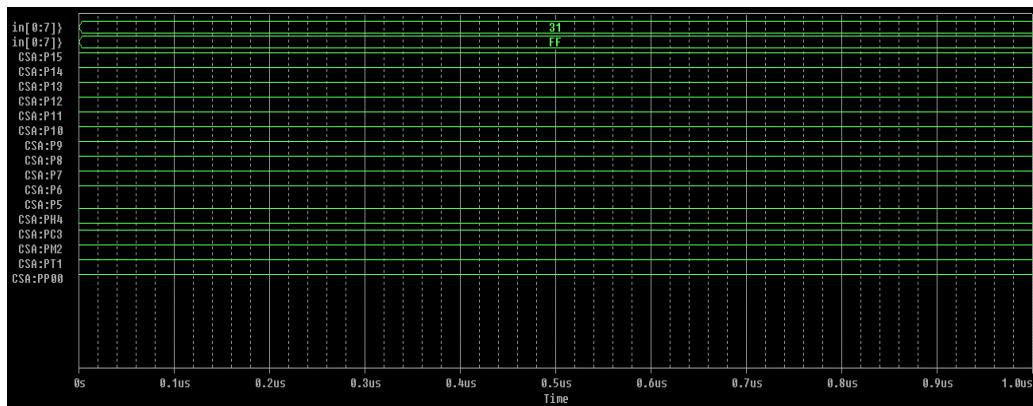
49 x 0 = 0:

Messages		
+ /wallace_mult/a	00110001	00110001
+ /wallace_mult/b	00000000	00000000
+ /wallace_mult/result	0000000000000000	0000000000000000



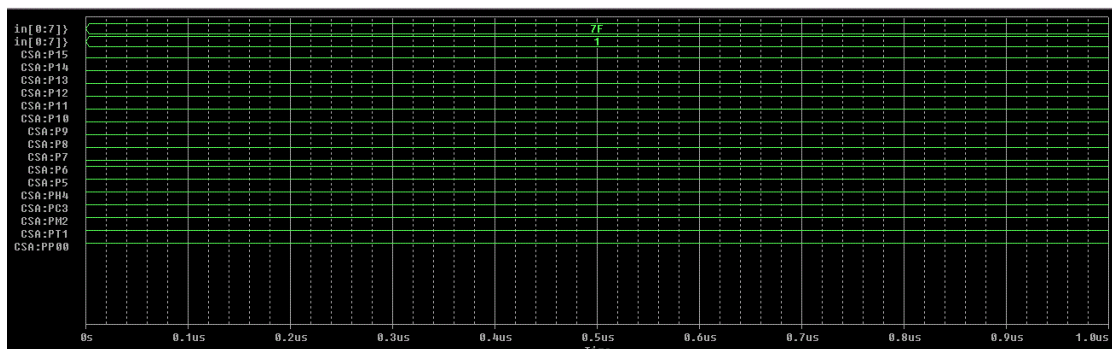
49 x -1 = -49:

Messages		
+ /wallace_mult/a	00110001	00110001
+ /wallace_mult/b	11111111	11111111
+ /wallace_mult/result	111111111001111	111111111001111



127 x 1 = 127:

Messages		
+ /wallace_mult/a	01111111	01111111
+ /wallace_mult/b	00000001	00000001
+ /wallace_mult/result	0000000001111111	0000000001111111



ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε επίπεδο λογικής οι full adders στο Wallace Tree είναι "στοιβαγμένοι" σε μια μεμονωμένη στήλη του πολλαπλασιαστή. Αυτό έχει ως αποτέλεσμα η στήλη να διατρέχεται από αγωγούς μεγάλου μήκους, χωρίς κανονικότητα, για την σύνδεση των

full adders. Η μεγάλη χωρητικότητα των αγωγών αυξάνει την καθυστέρηση και την κατανάλωση του πολλαπλασιαστή, ενώ επιπλέον δυσκολεύει τη διευθέτηση των αγωγών στο φυσικό σχέδιο. Παράλληλα όμως κερδίζουμε αρκετά σε ταχύτητα, αφού πλέον υπάρχουν μόνο $O(\log n)$ επίπεδα, με $O(1)$ propagation delay για το καθένα. Η παραγωγή των μερικών γινομένων είναι πολυπλοκότητας $O(1)$ και η τελική πρόσθεση $O(\log n)$, με αποτέλεσμα ο τελικός πολλαπλασιασμός να είναι πολυπλοκότητας $O(\log n)$ μόνο, δηλαδή ελάχιστα πιο αργός από την πρόσθεση.