

# Ψηφιακά Συστήματα ΗΥ σε Χαμηλά Επίπεδα Λογικής ΙΙ

Σχεδίαση και Υλοποίηση Πολλαπλασιαστή  
Κινητής Υποδιαστολής Μονής Ακριβείας  
(IEEE-754)

Ελένη Σούρλη

## Περιεχόμενα

0	Εισαγωγή	3
1	Main module	4
2	Normalization module	5
3	Rounding module	6
4	Exception Handling module	7
5	Testbench	8
6	System Verilog Assertions	9
7	Αποτελέσματα	10

## 0 Εισαγωγή

Η παρούσα εργασία επικεντρώνεται στην υλοποίηση ενός κυκλώματος πολλαπλασιασμού κινητής υποδιαστολής μονής ακρίβειας (32 – *bit*) σύμφωνα με το πρότυπο IEEE-754.

Σκοπός της εργασίας είναι η σχεδίαση του βασικού κυκλώματος του πολλαπλασιαστή, το οποίο περιλαμβάνει τη δημιουργία και διασύνδεση επιμέρους εσωτερικών modules, τα οποία επιτελούν συγκεκριμένες λειτουργίες όπως η εξαγωγή του προσήμου, ο χειρισμός του εκθέτη (*exponent handling*), ο πολλαπλασιασμός των σημαντικών ψηφίων (*mantissas*), η κανονικοποίηση (*normalization*), η στρογγυλοποίηση (*rounding*) και η διαχείριση ειδικών περιπτώσεων (*corner cases*).

Επιπλέον, πραγματοποιείται η δημιουργία διεξοδικού testbench σε SystemVerilog για την επαλήθευση της ορθότητας του συστήματος, τόσο για τυχαία όσο και για ακραία (*corner*) σενάρια. Τέλος, γίνεται χρήση SystemVerilog Assertions (SVAs), ώστε να διασφαλίζεται η ορθότητα των αποτελεσμάτων και η συνέπεια των παραγόμενων σημάτων κατά τη λειτουργία του κυκλώματος.

## 1 Main module

Το Main Module είναι υπεύθυνο για την εκτέλεση του βασικού υπολογισμού του πολλαπλασιασμού κινητής υποδιαστολής. Δέχεται δύο εισόδους των 32-bit ( $a$  και  $b$ ), καθώς και ένα σήμα λειτουργίας στρογγυλοποίησης 3-bit ( $rnd$ ), και παράγει ένα 32-bit αποτέλεσμα ( $z$ ) και ένα σήμα κατάστασης 8-bit ( $status$ ), το οποίο περιγράφει ειδικές συνθήκες όπως μηδενισμό, άπειρο, NaN, υπερχείλιση ή υποχείλιση.

Η λειτουργία του module ακολουθεί συγκεκριμένα στάδια. Αρχικά υπολογίζεται το πρόσημο του αποτελέσματος με XOR των προσήμων των εισόδων. Έπειτα αθροίζονται οι εκθέτες των δύο αριθμών και αφαιρείται το bias (127), ώστε να υπολογιστεί ο νέος εκθέτης. Στη συνέχεια γίνεται πολλαπλασιασμός των mantissas (με την προσθήκη του κρυφού bit), και το αποτέλεσμα κανονικοποιείται, δηλαδή ρυθμίζεται ώστε να είναι στη σωστή μορφή για αριθμούς κινητής υποδιαστολής.

Ακολουθεί η διέλευση από στάδιο pipeline για αύξηση της απόδοσης και αποφυγή καθυστερήσεων. Το pipeline στάδιο εφαρμόζεται μέσω καταχωρητών (pipeline registers), οι οποίοι ενεργοποιούνται με το θετικό μέτωπο του ρολογιού (posedge) και μηδενίζονται μέσω σήματος reset ενεργού-χαμηλού (active-low). Στη συνέχεια, εφαρμόζεται η επιλεγμένη μέθοδος στρογγυλοποίησης, βάσει του σήματος  $rnd$ . Τέλος, γίνεται χειρισμός εξαιρέσεων, όπως η εμφάνιση NaN, άπειρου ή μηδενικού αποτελέσματος, και καταγράφονται κατάλληλες τιμές (1, 0) στο status, ώστε να υποδεικνύεται το είδος του αποτελέσματος. Το module είναι σχεδιασμένο ώστε να είναι συμβατό με το πρότυπο IEEE-754 και αξιοποιεί αρχιτεκτονική pipeline για βελτιστοποίηση της απόδοσης.

## 2 Normalization module

Το normalization module είναι υπεύθυνο για την κανονικοποίηση της mantissa ενός αριθμού κινητής υποδιαστολής και για τον υπολογισμό των bit guard και sticky. Είσοδοι του module είναι το 48-bit αποτέλεσμα πολλαπλασιασμού P και το 10-bit άθροισμα των εκθετών, από το οποίο έχει αφαιρεθεί το bias. Έξοδοι είναι τα bits guard και sticky, η 23-bit κανονικοποιημένη mantissa και ο 10-bit κανονικοποιημένος εκθέτης.

Η θέση του δυαδικού σημείου βρίσκεται μεταξύ των bits P[46] και P[45]. Αν το MSB του αποτελέσματος πολλαπλασιασμού είναι ίσο με 1, σημαίνει ότι ο αριθμός πριν το δυαδικό σημείο είναι της μορφής '10' ή '11', οπότε γίνεται αριστερή μετατόπιση της mantissa και αύξηση του εκθέτη κατά ένα. Αν το MSB είναι 0, τότε ο αριθμός είναι της μορφής '01', δεν απαιτείται μετατόπιση, και ο εκθέτης παραμένει σταθερός. Στην περίπτωση αυτή, το P[46] θεωρείται ως το leading one.

Το sticky bit προκύπτει από το λογικό OR όλων των υπολοίπων bits της 48-bit mantissa μετά τα guard bits. Οι έξοδοι του module χρησιμοποιούνται στο επόμενο στάδιο για τη στρογγυλοποίηση του αριθμού.

### 3 Rounding module

Το rounding module είναι υπεύθυνο για τη στρογγυλοποίηση της mantissa μετά την κανονικοποίηση, σύμφωνα με την είσοδο ελέγχου round, η οποία καθορίζει τον τύπο της στρογγυλοποίησης που θα εφαρμοστεί. Το module λαμβάνει ως είσοδο μια 24-bit mantissa (συμπεριλαμβανομένου του leading one), καθώς και τα guard και sticky bits, τα οποία χρησιμοποιούνται τόσο για την απόφαση στρογγυλοποίησης όσο και για τον προσδιορισμό της ακρίβειας του αποτελέσματος. Αν και τα δύο bits είναι μηδενικά, το αποτέλεσμα θεωρείται ακριβές και το σήμα inexact τίθεται σε 0· διαφορετικά, το inexact λαμβάνει τιμή 1.

Η έξοδος του module είναι ένα 25-bit σήμα, ώστε να υποστηρίζεται πιθανή υπερχείλιση κατά τη στρογγυλοποίηση. Σε περίπτωση άκυρης τιμής της round, εφαρμόζεται προεπιλεγμένα η στρογγυλοποίηση τύπου IEEE\_near. Εάν το πιο σημαντικό bit της 25-bit mantissa είναι 1 μετά τη στρογγυλοποίηση, τότε εφαρμόζεται δεξιά μετατόπιση κατά ένα bit και αύξηση του εκθέτη κατά ένα (post-rounding normalization). Αν δεν υπάρξει υπερχείλιση, η mantissa και ο εκθέτης παραμένουν αμετάβλητοι.

Η τελική 24-bit mantissa και ο post-normalized εκθέτης συνδυάζονται για να παραχθεί το σήμα z\_calc, το οποίο έχει τη μορφή της τελικής εξόδου z, αποτελούμενο από 1 bit πρόσημο, 8 bits εκθέτη και 23 bits mantissa, και χρησιμοποιείται στο στάδιο διαχείρισης εξαιρέσεων.

## 4 Exception Handling module

Το exception handling module είναι υπεύθυνο για τη διαχείριση ακραίων περιπτώσεων, όπως η υπερχείλιση από τον πολλαπλασιασμό δύο μεγάλων αριθμών ή ο πολλαπλασιασμός απειρισμένων τιμών. Είσοδοι του module είναι τα αρχικά 32-bit σήματα `a` και `b`, το 32-bit αποτέλεσμα μετά τη στρογγυλοποίηση `z_calc`, τα bits `overflow`, `underflow`, και `inexact`, καθώς και η είσοδος `round`.

Τα σήματα `overflow` και `underflow` υπολογίζονται στο κύριο module μετά το στάδιο στρογγυλοποίησης. Υπερχείλιση συμβαίνει όταν ο εκθέτης μετά τη στρογγυλοποίηση ξεπερνά το μέγιστο επιτρεπτό όριο, ενώ υποχείλιση όταν είναι μικρότερος από το ελάχιστο. Έξοδοι του module είναι το 32-bit σήμα `z` και έξι 1-bit σήματα κατάστασης: `zero_f`, `inf_f`, `nan_f`, `tiny_f`, `huge_f`, `inexact_f`, τα οποία συνδυάζονται στο κύριο module για τη δημιουργία του σήματος `status`.

Εσωτερικά, ορίζεται ένας enum type με όνομα `interp_t` που περιλαμβάνει τις κατηγορίες τιμών κινητής υποδιαστολής: `ZERO`, `INF`, `NORM`, `MIN_NORM`, `MAX_NORM`. Οι περιπτώσεις NaNs και Denormals αντιμετωπίζονται ως `INF` και `ZERO` αντίστοιχα.

Ορίζονται δύο βοηθητικές συναρτήσεις: `num_interp`, που λαμβάνει έναν 32-bit αριθμό και επιστρέφει την κατηγορία του ως `interp_t`, και η `z_num`, που κάνει το αντίστροφο, επιστρέφοντας τη δυαδική αναπαράσταση της τιμής βάσει του τύπου.

Τέλος, σε ένα μπλοκ `always_comb`, αρχικοποιούνται τα σήματα κατάστασης σε μηδέν, και με βάση τον συνδυασμό εισόδων και τις συναρτήσεις, υπολογίζεται η τελική έξοδος `z` και ενεργοποιούνται τα κατάλληλα bits κατάστασης. Στην περίπτωση `Norm × Norm`, αν υπάρξει `overflow`, το αποτέλεσμα στρογγυλοποιείται είτε σε `maxNormal` είτε σε `INF`, ανάλογα με την είσοδο `round` και το πρόσημο του `z_calc`. Σε περίπτωση `underflow`, το αποτέλεσμα γίνεται `minNormal` ή `ZERO`. Αν δεν υπάρξει κάποια από τις δύο περιπτώσεις, τότε το `z` είναι ίσο με το `z_calc`, και το `inexact_f` αντιγράφει την τιμή του `inexact`.

## 5 Testbench

Το testbench (fpmulttb) σχεδιάστηκε με στόχο την πλήρη επαλήθευση της λειτουργικότητας του πολλαπλασιαστή κινητής υποδιαστολής, σύμφωνα με το πρότυπο IEEE-754. Το testbench περιλαμβάνει τόσο τυχαίες δοκιμές όσο και δοκιμές γωνιακών περιπτώσεων (corner cases), προκειμένου να διασφαλιστεί η ακρίβεια και η αξιοπιστία της υλοποίησης σε όλες τις πιθανές εισόδους.

Δημιουργείται σήμα με περίοδο 10ns, ενώ για κάθε τρόπο στρογγυλοποίησης εκτελούνται 60 δοκιμές με τυχαίες τιμές εισόδου a και b, οι οποίες παράγονται μέσω της \$urandom(). Οι τρόποι στρογγυλοποίησης που ελέγχονται είναι οι εξής: `IEEE_near`, `IEEE_zero`, `IEEE_pinf`, `IEEE_ninf`, `near_up` και `away_zero`. Για κάθε περίπτωση, το αναμενόμενο αποτέλεσμα υπολογίζεται μέσω της συνάρτησης `multiplication`, η οποία ενσωματώνει τη σωστή λογική στρογγυλοποίησης. Έπειτα, η έξοδος z του DUT συγκρίνεται με το `real_z` μετά από τρεις κύκλους και σε περίπτωση αναντιστοιχίας καταγράφεται σφάλμα.

Παράλληλα, το testbench περιλαμβάνει και δοκιμές για γωνιακές περιπτώσεις (corner cases), εξετάζοντας όλους τους δυνατούς συνδυασμούς μεταξύ ειδικών αριθμητικών τιμών όπως `signaling` και `quiet NaN` (θετικά και αρνητικά), θετικό και αρνητικό άπειρο, κανονικοποιημένοι και υποκανονικοποιημένοι αριθμοί, καθώς και θετικά και αρνητικά μηδενικά. Οι περιπτώσεις αυτές αξιολογούνται και στους έξι τρόπους στρογγυλοποίησης.

Τέλος, το testbench καταγράφει τις επιτυχίες και τις αποτυχίες τόσο για τις τυχαίες όσο και για τις γωνιακές περιπτώσεις, προβάλλοντας αναλυτικά αποτελέσματα μετρικών για κάθε σενάριο.



## 6 System Verilog Assertions

Η ενότητα αυτή αφορά την υλοποίηση άμεσων και συγχρονισμένων λογικών ελέγχων (Immediate και Concurrent Assertions) για τη διασφάλιση της λειτουργικής ορθότητας του πολλαπλασιαστή κινητής υποδιαστολής. Η υλοποίηση χωρίζεται σε δύο επιμέρους υποενότητες.

Στο πρώτο μέρος, δημιουργείται η μονάδα `test_status_bits` η οποία περιλαμβάνει όλους τους δυνατούς άμεσους λογικούς ελέγχους μεταξύ των `status bits`, ώστε να διασφαλιστεί ότι δεν υπάρχουν ασύμβατοι συνδυασμοί. Κάποια ζεύγη σηματοδοτήσεων δεν πρέπει να ενεργοποιούνται ταυτόχρονα. Οι έλεγχοι αυτοί υλοποιούνται με τη χρήση SystemVerilog immediate assertions και επαληθεύουν ότι κάθε απαγορευμένος συνδυασμός δεν εμφανίζεται.

Στο δεύτερο μέρος, δημιουργείται η μονάδα `test_status_z_combinations` η οποία περιλαμβάνει πέντε συγχρονισμένους ελέγχους (concurrent assertions) που ενεργοποιούνται σε κάθε θετικό παλμό του ρολογιού. Οι έλεγχοι αυτοί επαληθεύουν ότι η τιμή των `bit` του πεδίου `exchété` και της σημαντικής του σήματος εξόδου `z` ανταποκρίνεται σωστά στην τιμή των αντίστοιχων `status bits`. Συγκεκριμένα, επαληθεύεται ότι:

- Αν το `bit zero` είναι 1, τότε όλα τα `bit` του `exchété` του `z` είναι 0.
- Αν το `bit inf` είναι 1, τότε όλα τα `bit` του `exchété` του `z` είναι 1.
- Αν το `bit nan` είναι 1, τότε τρεις κύκλους νωρίτερα, το πεδίο `exchété` των εισόδων `a` και `b` πρέπει να έχουν μεταξύ τους το μοτίβο (000... , 111...) ή το αντίστροφο.
- Αν το `bit huge` είναι 1, τότε το πεδίο `exchété` του `z` είναι είτε πλήρως 1 (infinity ) είτε ισούται με την τιμή του `maxNormal` (όλα 1 εκτός από το LSB και mantissa γεμάτη με 1).
- Αν το `bit tiny` είναι 1, τότε το πεδίο `exchété` του `z` είναι είτε πλήρως 0 (zero ) είτε ισούται με την τιμή του `minNormal` (όλα 0 εκτός από το LSB και mantissa γεμάτη με 0).

Οι δύο παραπάνω μονάδες συνδέονται με το `wrapper module` μέσω της διαδικασίας `bind`, ώστε να διασφαλιστεί η αυτόματη επαλήθευση της λειτουργίας κατά τη διάρκεια προσομοιώσεων.

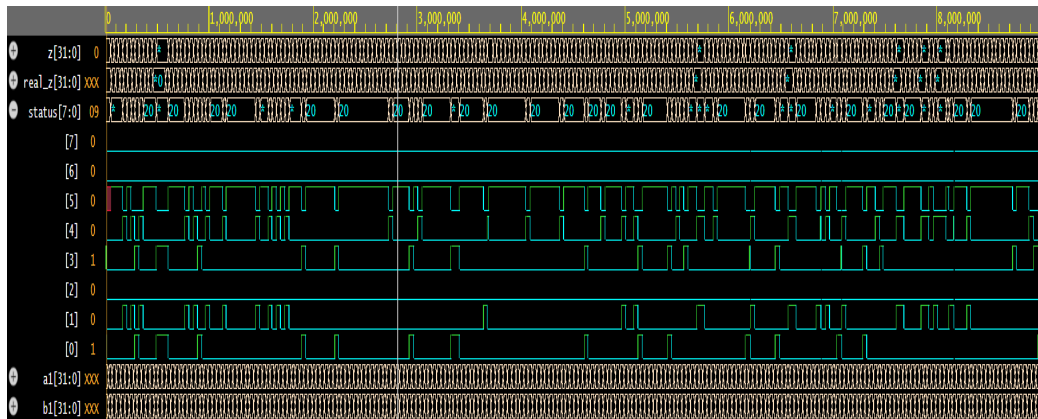
## 7 Αποτελέσματα

```
# run -all
# + PASSED at IEEE_near for 60 tests
# + PASSED at IEEE_zero for 60 tests
# + PASSED at IEEE_pinf for 60 tests
# + PASSED at IEEE_ninf for 60 tests
# + PASSED at near_up for 60 tests
# + PASSED at away_zero for 60 tests
# === Starting Corner Case Tests ===
# === Completed Corner Case Tests ===
# Simulation complete.
# Successful comparisons: 864
# Failed comparisons:    0
# Total comparisons:    864
# ** Note: $stop      : testbench.sv(203)
# Time: 49975 ns  Iteration: 0  Instance: /fp_mult_tb
```

Σχήμα 1: αποτελέσματα δοκιμών για όλες τις περιπτώσεις

Τα αποτελέσματα της προσομοίωσης καταδεικνύουν την πλήρη επιτυχία του συστήματος στις διάφορες ρυθμίσεις στρογγυλοποίησης σύμφωνα με το πρότυπο IEEE, με 60 επιτυχείς δοκιμές σε κάθε περίπτωση. Επιπλέον, κατά την εκτέλεση των Corner Case Tests, δεν παρατηρήθηκε καμία αποτυχία, γεγονός που επιβεβαιώνεται από το σύνολο των 864 επιτυχημένων συγκρίσεων έναντι μηδενικών αποτυχημένων. Το αποτέλεσμα αυτό αποδεικνύει την ορθότητα και την αξιοπιστία της υλοποίησης υπό όλες τις συνθήκες στρογγυλοποίησης και οριακών περιπτώσεων. Τέλος, δεν παρατηρείται η παραβίαση κάποιου assertion .

Δίνεται ενδεικτικά το waveformορισμένων αποτελεσμάτων για κάποιες τυχαίες τιμές των  $a$ ,  $b$ . Επιβεβαιώνεται και με αυτόν τον τρόπο η ταύτιση των αποτελεσμάτων του πολλαπλασιασμού και της δοθείσας συνάρτησης `multiplication`.



Σχήμα 2: Προσομοίωση τυχαίων τιμών  $a$ ,  $b$