



# **Υπολογιστικά Συστήματα Υψηλής Αξιοπιστίας**

## **Εγχειρίδιο χρήσης Hore 2.0**

Επιμέλεια: Δρ. Γκάμας Βασίλειος

Πανεπιστήμιο Δυτικής Αττικής  
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

## ΠΕΡΙΕΧΟΜΕΝΑ

1	Εισαγωγή.....	3
2	Εκτέλεση σε περιβάλλον Windows (DOS).....	3
3	Ορίσματα εκτέλεσης Hope .....	3
4	Αρχεία κυκλωμάτων (benchmarks) .....	4
5	Αρχείο προτύπων δοκιμής (test patterns).....	6
6	Αρχείο λίστας σφαλμάτων (faults list) .....	6
7	Παραδείγματα χρήσης του Hope.....	7

## 1 Εισαγωγή

Το Hope αποτελεί έναν Παράλληλο Εξομοιωτή Σφαλμάτων (Fault Simulator) για σύγχρονα ακολουθιακά κυκλώματα. Χρησιμοποιεί την τεχνική παράλληλης προσομοίωσης σφαλμάτων και διάφορα heuristics για την μείωση του χρόνου προσομοίωσης σφαλμάτων. Το λογισμικό Hope μπορεί να χρησιμοποιηθεί τόσο σε περιβάλλον Windows (DOS) όσο και σε περιβάλλον Unix/Linux με μικρές διαφοροποιήσεις όσον αφορά τα testbenches.

Το παρόν εγχειρίδιο έχει ως σκοπό να σας καθοδηγήσει στην χρήση του εργαλείου σε περιβάλλον Windows (DOS).

## 2 Εκτέλεση σε περιβάλλον Windows (DOS)

Για να τρέξετε το Hope σε περιβάλλον Windows (DOS) εκτελέστε τα παρακάτω βήματα:

1. Αντιγράψτε τον φάκελο Hope που περιέχει το εκτελέσιμο αρχείο και τον πηγαίο κώδικα του Hope σε έναν φάκελο στον σκληρό δίσκο του υπολογιστή σας. Τα αρχεία που σας δίνονται έχουν ήδη μετατραπεί σε κατάλληλη μορφή ώστε να μπορέσετε να εκτελέσετε το Hope σε περιβάλλον Windows.
2. Αντιγράψτε τα αρχεία κυκλωμάτων (\*.bench) που βρίσκονται στον φάκελο Benchmarks μέσα στον φάκελο Hope ώστε κατά την εκτέλεση του προγράμματος να μην εμφανιστεί σφάλμα μη εύρεσης των κυκλωμάτων.
3. Ανοίξτε το command prompt στα Windows πληκτρολογώντας στην κονσόλα εντολών την εντολή **cmd**.
4. Μετακινηθείτε στον φάκελο Hope χρησιμοποιώντας την εντολή **cd**. Για παράδειγμα αν το Hope βρίσκεται στην επιφάνεια εργασίας εκτελέστε την εντολή **cd Desktop\Hope**. Εκτελώντας την εντολή **dir** μπορείτε να δείτε τα περιεχόμενα του φακέλου, το εκτελέσιμο αρχείο του Hope (.exe), τα αρχεία των κυκλωμάτων (.bench), κτλ.

Για να τρέξετε το Hope εκτελέστε την παρακάτω εντολή:

**hope [options] circuit\_file [>outfile]**

όπου,

- **options**: ορίσματα σχετικά με την παραγωγή των επιθυμητών αποτελεσμάτων (προαιρετική παράμετρος)
- **circuit\_file**: αρχείο κυκλώματος (υποχρεωτική παράμετρος)
- **outfile**: αρχείο εξόδου στο οποίο επιθυμούμε να καταγραφούν τα δεδομένα που προκύπτουν από την εκτέλεση του προγράμματος (προαιρετική παράμετρος)

## 3 Ορίσματα εκτέλεσης Hope

Παρακάτω παρατίθενται τα διάφορα ορίσματα εκτέλεσης του Hope.

-c fn	Options are read from the file "fn". (default: online command mode)
-D	The list of faults which are newly detected by a test pattern is reported in the log file. The option -l should be specified. (default: only the number of faults detected is reported.)
-f fn	Faults are defined in the file "fn". (default: faults are generated internally)
-F fn	The good and faulty circuit outputs are reported for each fault in the file fn. In this option, the fault dropping heuristic of HOPE is not performed, i.e., all the faults are injected and simulated in parallel. (default: faulty circuit output is not reported.)

-l fn	A log file named "fn" is created.
-------	-----------------------------------

Επιλογές στη διαδικασία εξομίωσης.

-N	Diagnostic mode. No fault dropping is performed. That is, all the faults are simulated for each test pattern. (default: faults detected during the fault simulation are dropped from the fault list.)
-r n	(Random pattern generation mode). Test patterns are generated randomly. The fault simulation stops either when all faults are detected or n patterns are applied. (default: -r 224)
-s n	Initial seed for the random number generator is set by n. If n=0, random seed is generated using the day time of the computer. (default: -s 0)
-t fn	Test patterns are provided in the file "fn". (default: random patterns are used)
-u	Prints out all undetected faults in a file. The name of this file is <ckt>.ufaults. Note that hope does not update a fault file if one already exists in the run directory. This fault list file can be directly read by atalanta or hope. (default: no file is created)
-U fn	The same as -u, but hope writes undetected faults to the given file name. (default: no file is created)
-x	(Potential detection mode). Potentially detected faults are dropped as soon as they are detected. (default: only detected faults are dropped)
-0	All the flip-flops are initially set to logic 0
-1	All the flip-flops are initially set to logic 1. (default: All the flip-flops are initially set to unknown (X).)

Τέλος υφίστανται και τα παρακάτω ορίσματα.

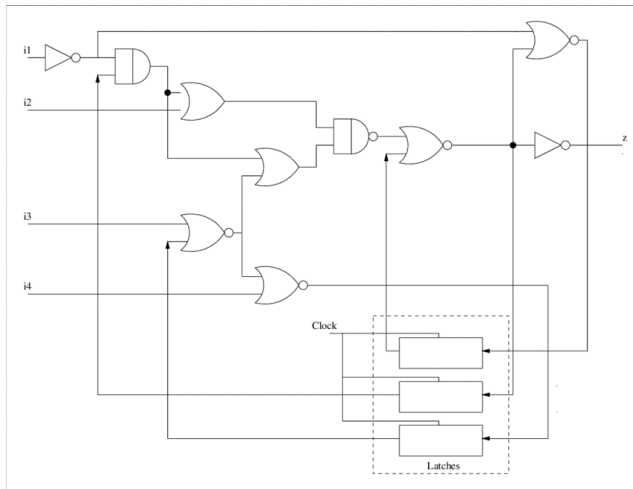
-h f	Displays an example fault list format.
-h g	Displays the on-line user's guide.
-h n	Displays an example netlist format.
-h t	Displays an example test pattern file.
-h a	Displays the entire on-line manual file. (default: no manual is displayed)

Να σημειώσουμε ότι στο default mode, δε δημιουργείται κανένα αρχείο εξόδου. Η περίληψη της εξομίωσης σφαλμάτων καταγράφεται στην έξοδο εκτέλεσης του Hope στο περιβάλλον του DOS. Αν έχει οριστεί η επιλογή -l τότε δημιουργείται ένα log αρχείο, το οποίο περιέχει περισσότερες πληροφορίες για τη διαδικασία εξομίωσης σφαλμάτων.

## 4 Αρχεία κυκλωμάτων (benchmarks)

Η μορφή του netlist για το Hope είναι σε γενικές γραμμές τα ISCAS89 με κάποιες μικρές αλλαγές. Η δομή ενός benchmark αρχείου φαίνεται παρακάτω για ένα ενδεικτικό λογικό κύκλωμα (s27). Οι γραμμές οι οποίες ξεκινούν με δίεση (#) περιλαμβάνουν σχόλια και δε λαμβάνονται υπόψιν κατά την επεξεργασία του αρχείου. Η πρώτη γραμμή περιλαμβάνει το όνομα του λογικού κυκλώματος και οι ακόλουθες γραμμές με σχόλια περιλαμβάνουν το πλήθος των εισόδων, εξόδων, flip flop, αντιστροφέων και πυλών του κυκλώματος. Η σειρά με την οποία εμφανίζονται

οι λογικές πύλες μέσα στο αρχείο δεν παίζει κάποιο ρόλο. Τέλος, τα ονόματα των πυλών πρέπει να είναι αλφαριθμητικοί χαρακτήρες (0-9, A-Z, a-z, \_, [ ή ]).



# s27  
 # 4 inputs  
 # 1 outputs  
 # 3 D-type flipflops  
 # 2 inverters  
 # 8 gates (1 ANDs + 1 NANDs +  
 2 ORs + 4 NORs)

INPUT(G0)  
 INPUT(G1)  
 INPUT(G2)  
 INPUT(G3)

OUTPUT(G17)

G5 = DFF(G10)  
 G6 = DFF(G11)  
 G7 = DFF(G13)

G14 = NOT(G0)  
 G17 = NOT(G11)

G8 = AND(G14, G6)

G15 = OR(G12, G8)  
 G16 = OR(G3, G8)

G9 = NAND(G16, G15)

G10 = NOR(G14, G11)  
 G11 = NOR(G5, G9)  
 G12 = NOR(G1, G7)  
 G13 = NOR(G2, G12)

Οι πύλες που υποστηρίζονται είναι οι παρακάτω:

Τύπος στοιχείου	Εντολή
Κύρια είσοδος	INPUT
Κύρια έξοδος	OUTPUT
Πύλη AND	AND
Πύλη NAND	NAND
Πύλη OR	OR
Πύλη NOR	NOR
Πύλη XOR 2 εισόδων	XOR
Buffer	BUFF ή BUF
Αντιστροφέας	NOT

## 5 Αρχείο προτύπων δοκιμής (test patterns)

Κάθε γραμμή του αρχείου η οποία ξεκινάει με \* αποτελεί σχόλιο και αγνοείται κατά την επεξεργασία του αρχείου. Κάθε πρότυπο δοκιμής ξεκινάει με το σύμβολο : . Στην περίπτωση ενός λογικού κυκλώματος η εισόδων, μόνο τα η bits που ακολουθούν το : είναι σημαντικά και τα υπόλοιπα αγνοούνται. Παρακάτω παρατίθεται το αρχείο test pattern για το λογικό κύκλωμα s27 (s27.test). Υπάρχουν 10 πρότυπα δοκιμής. Για παράδειγμα το πρώτο πρότυπο δοκιμής είναι το 0110 . Αυτό σημαίνει ότι οι 4 εισοδοι του κυκλώματος θα λάβουν τις παρακάτω λογικές τιμές:

- Είσοδος G0: 0
- Είσοδος G1: 1
- Είσοδος G2: 1
- Είσοδος G3: 0

```
* Name of circuit: s27
```

```
1: 0110
2: 1111
3: 0110
4: 1011
5: 1110
6: 0001
7: 0100
8: 1011
9: 0011
10: 0011
```

## 6 Αρχείο λίστας σφαλμάτων (faults list)

Υπάρχει η δυνατότητα να δώσει ο χρήστης στο Hore την λίστα των σφαλμάτων τα οποία επιθυμεί να εξομοιωθούν. Όταν χρησιμοποιείται η επιλογή "-f fn" η λίστα σφαλμάτων διαβάζεται από το αρχείο fn. Παρακάτω δίνεται ένα παράδειγμα ενός αρχείου λίστας σφαλμάτων.

```
gate_A->gate_B /1
gate_A->gate_B /0
gate_A /1
gate_B /1
```

Οι εντολές gate\_A και gate\_B ορίζουν τα ονόματα των πυλών. Η πρώτη γραμμή, gate\_A->gate\_B /1 περιγράφει stuck-at 1 σφάλμα στην πύλη B που είναι συνδεδεμένη με την πύλη A. Παρόμοια, η δεύτερη γραμμή περιγράφει stuck-at 0 σφάλμα στη πύλη A που είναι συνδεδεμένη με την πύλη B. Η τρίτη και η τέταρτη γραμμή περιγράφουν stuck-at 1 σφάλματα στις εξόδους των πυλών A και B αντίστοιχα.

Να σημειωθεί ότι δεν εφαρμόζεται fault collapsing με αυτή την επιλογή. Γι' αυτό το λόγο, οι χρήστες θα πρέπει να παρέχουν τη λίστα σφαλμάτων η οποία περιέχει και τα collapsed σφάλματα.

## 7 Παραδείγματα χρήσης του Hope

Παρακάτω παρατίθενται διάφορα παραδείγματα χρήσης του Hope.

### Παράδειγμα #1

```
>hope -D s27.bench
```

Προσομοιώνεται το κύκλωμα s27.bench. Η λίστα των σφαλμάτων που ανιχνεύονται από κάθε πρότυπο δοκιμής αναφέρεται σε ένα αρχείο log που δημιουργείται.

```
C:\Users\user\Desktop\Hope>hope -D s27.bench
*****
*
*           Welcome to HOPE (version 2.0)
*
*           Dong S. Ha (ha@vt.edu)
*           Web: http://www.visc.vt.edu/people/ha
*           Virginia Polytechnic Institute & State University
*
*****

*****  SUMMARY OF SIMULATION RESULTS  *****
1. Circuit structure
   Name of circuit           : s27____
   Number of primary inputs  : 4
   Number of primary outputs : 1
   Number of flip-flops     : 3
   Number of gates          : 10
   Level of the circuit     : 6

2. Simulator input parameters
   Simulation mode           : random
   Initial random number generator seed : 1605712766

3. Simulation results
   Number of test patterns applied : 224
   Fault coverage              : 100.000 %
   Number of collapsed faults   : 32
   Number of detected faults   : 32
   Number of undetected faults : 0

4. Memory used                : 167862 Kbytes

5. CPU time
   Initialization            : 1816659.367 secs
   Fault simulation          : 0.000 secs
   Total                     : 1816659.367 secs
```

## Παράδειγμα #2

```
>hope -t s27.test s27.bench
```

Προσομοιώνεται το κύκλωμα s27.bench χρησιμοποιώντας τα πρότυπα δοκιμής στο αρχείο "s27.test". Η προσομοίωση σταματάει όταν προσομοιωθούν όλα τα πρότυπα δοκιμής που περιέχονται στο αρχείο "s27.test" ή ανιχνευθούν όλα τα σφάλματα.

```
C:\Users\user\Desktop\Hope>hope -t s27.test s27.bench
*****
*
*           Welcome to HOPE (version 2.0)
*
*           Dong S. Ha (ha@vt.edu)
*           Web: http://www.visc.vt.edu/people/ha
*           Virginia Polytechnic Institute & State University
*
*****

*****  SUMMARY OF SIMULATION RESULTS  *****
1. Circuit structure
   Name of circuit           : s27____
   Number of primary inputs  : 4
   Number of primary outputs : 1
   Number of flip-flops      : 3
   Number of gates           : 10
   Level of the circuit      : 6

2. Simulator input parameters
   Simulation mode           : file (s27.test)

3. Simulation results
   Number of test patterns applied : 10
   Fault coverage              : 78.125 %
   Number of collapsed faults    : 32
   Number of detected faults     : 25
   Number of undetected faults   : 7

4. Memory used                : 167862 Kbytes

5. CPU time
   Initialization             : 1805878.633 secs
   Fault simulation            : 0.000 secs
   Total                       : 1805878.633 secs
```



### Παράδειγμα #3

```
> hope -s 9999 -r 20000 s27.bench > s27.out
```

Προσομοιώνεται το κύκλωμα s27.bench χρησιμοποιώντας 20000 τυχαία πρότυπα δοκιμής. Η γεννήτρια τυχαίων αριθμών αρχικοποιείται με τη τιμή 9999. Η προσομοίωση σφαλμάτων ολοκληρώνεται όταν 20000 τυχαία πρότυπα δοκιμής προσομοιώνονται ή όταν ανιχνευθούν όλα τα σφάλματα. Η έξοδος εκτέλεσης της εντολής αποθηκεύεται στο αρχείο "s27.out".

### Παράδειγμα #4

```
> hope -s 9999 -r 20000 s27.bench -l s27.log
```

Προσομοιώνεται το κύκλωμα s27.bench χρησιμοποιώντας 20000 τυχαία πρότυπα δοκιμής. Η γεννήτρια τυχαίων αριθμών αρχικοποιείται με τη τιμή 9999. Η προσομοίωση σφαλμάτων ολοκληρώνεται όταν 20000 τυχαία πρότυπα δοκιμής προσομοιώνονται ή όταν ανιχνευθούν όλα τα σφάλματα. Παράγεται ένα log αρχείο (s27.log) που περιέχει πληροφορίες σχετικά με το πλήθος σφαλμάτων που ανιχνεύεται από κάθε διάνυσμα δοκιμής καθώς και επιπλέον πληροφορίες.

```
C:\Users\user\Desktop\Hope>hope -s 9999 -r 20000 s27.bench -l s27.log
*****
*
*           Welcome to HOPE (version 2.0)           *
*
*           Dong S. Ha (ha@vt.edu)                 *
*           Web: http://www.visc.vt.edu/people/ha   *
*           Virginia Polytechnic Institute & State University *
*
*****

*****  SUMMARY OF SIMULATION RESULTS  *****
1. Circuit structure
   Name of circuit           : s27____
   Number of primary inputs  : 4
   Number of primary outputs : 1
   Number of flip-flops      : 3
   Number of gates           : 10
   Level of the circuit      : 6

2. Simulator input parameters
   Simulation mode           : random
   Initial random number generator seed : 9999

3. Simulation results
   Number of test patterns applied : 20000
   Fault coverage                : 100.000 %
   Number of collapsed faults    : 32
   Number of detected faults     : 32
   Number of undetected faults   : 0

4. Memory used                : 167862 Kbytes

5. CPU time
   Initialization             : 1811088.533 secs
   Fault simulation            : 0.517 secs
   Total                       : 1811089.050 secs
```

## Παράδειγμα #5

```
>hope -D 3 -U c432o.faults c432o.bench
```

Παράγονται 3 διανύσματα δοκιμής για κάθε σφάλμα και δημιουργείται ένα αρχείο με όνομα c432o.faults στο οποίο αποθηκεύονται τα σφάλματα που αποβάλλονται (aborted). Αν δεν υπάρχουν τέτοια σφάλματα (αποτυπώνεται στις πληροφορίες που επιστρέφονται με το τέλος εκτέλεσης της εντολής), δε δημιουργείται το σχετικό αρχείο.

```
C:\Users\user\Desktop\Atalanta>atalanta -D 3 -U c432o.faults c432o.bench
*****
*
*      Welcome to atalanta (version 2.0)      *
*
*      Dong S. Ha (ha@vt.edu)                 *
*      Web: http://www.visc.vt.edu/people/ha   *
*      Virginia Polytechnic Institute & State University *
*
*****
*****  SUMMARY OF TEST PATTERN GENERATION RESULTS  *****
1. Circuit structure
   Name of the circuit           : c432o
   Number of primary inputs      : 36
   Number of primary outputs     : 7
   Number of gates               : 160
   Level of the circuit          : 17
2. ATPG parameters
   Test pattern generation mode   : DTPG + TC
   Backtrack limit               : 10
   Initial random number generator seed : 1602863234
   Test pattern compaction mode   : NONE
3. Test pattern generation results
   Number of test patterns        : 1406
   Fault coverage                 : 98.855 %
   Number of collapsed faults    : 524
   Number of identified redundant faults : 1
   Number of aborted faults      : 5
   Total number of backtrackings  : 1075
4. Memory used                   : 167964 Kbytes
5. CPU time
   Initialization                 : 0.000 secs
   Fault simulation                : 0.000 secs
   FAN                            : 0.250 secs
   Total                          : 0.250 secs
```