

Υπολογισμός stress-tests με χρήση γενετικών αλγορίθμων - Υπολογισμός φόρτου εργασίας μέγιστης κατανάλωσης ισχύος

Vasileios Tenentes

University of Ioannina

Έχουμε τελειώσει με τη δυναμική κατανάλωση;

Μπορούμε να υπολογίσουμε σε ένα λογικό σχεδιασμό:

- το switching activity γνωστού φόρτου εργασίας
- το **average switching activity** άγνωστου φόρτου εργασίας
 - Μέσω Monte Carlo
 - Μέσω Signal Probabilities

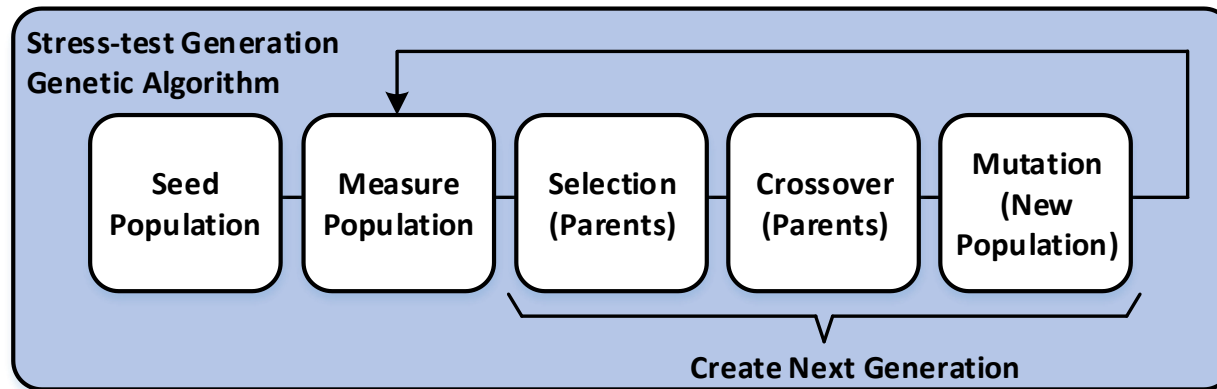
Πολλές φορές όμως είναι πρακτικό να ξέρουμε το **μέγιστο switching activity**, λέγεται και worst switching activity, ενός κυκλώματος. Πως μπορούμε να το υπολογίσουμε;

Υπολογισμός μέγιστης δυναμικής κατανάλωσης

Δεν είναι εύκολος στόχος γιατί:

Χρειαζόμαστε έναν συστηματικό τρόπο να εντοπίσουμε το **workload** που προκαλεί τη **μέγιστη δυναμική κατανάλωση ισχύος**. Συγκεκριμένα αυτό το workload ονομάζεται **power stress-test**.

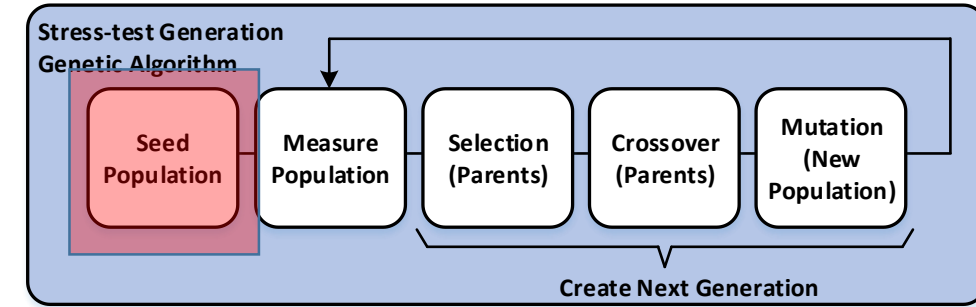
Η αυτοματοποιημένη μεθοδολογία παραγωγής stress-tests βασίζεται σε γενετικούς αλγορίθμους.



Η τεχνική μπορεί να εφαρμοστεί σε επεξεργαστές και λογικά κυκλώματα.

Βήμα 1^ο: Αρχικός πληθυσμός (σπόρος – seed)

Υπενθυμίζουμε ότι το workload είναι μια χρονοσειρά από τιμές στις εισόδους του κυκλώματος.

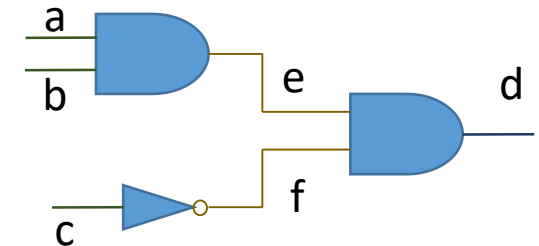


Παράμετροι του αλγορίθμου

- **Μήκος φόρτου εργασίας L (workload length)**: είναι το μέγεθος μιας χρονοσειράς και το μετράμε σε **πλήθος διανυσμάτων εισόδου**.
- **Μέγεθος πληθυσμού N (population size)**: το πλήθος των ξεχωριστών φόρτων εργασίας που θα εξερευνάει ο αλγόριθμος σε κάθε του βήμα.

Αρχικά ξεκινάμε με έναν πληθυσμό από N τυχαία workloads. Κάθε ξεχωριστό workload (**individual workload**) είναι μια χρονοσειρά μήκους L.

π.χ. αρχικός πληθυσμός για L=3, N=4



Individual workload 1

t	a	b	c
t1	1	0	1
t2	0	0	1
t3	1	1	0

Individual workload 2

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Individual workload 3

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

Individual workload 4

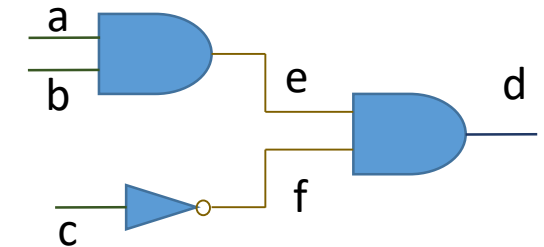
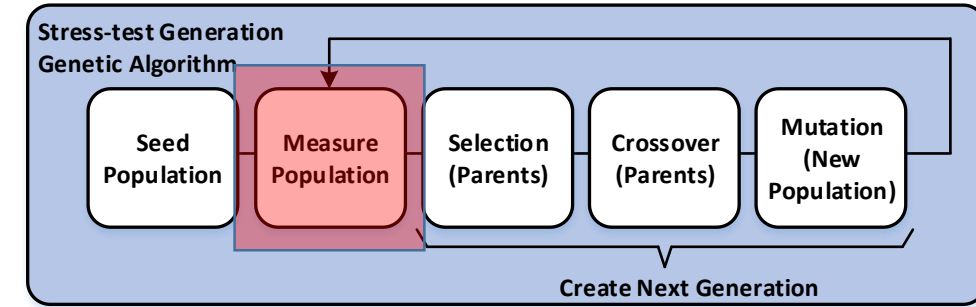
t	a	b	c
t1	1	1	1
t2	0	0	0
t3	1	0	1

Βήμα 2^ο: Μέτρηση της κατανάλωσης κάθε individual

Πρέπει να μετρήσουμε πόσο ένας individual πετυχαίνει **τον στόχο**.
ΜΕΤΡΑΜΕ ΤΟΝ ΣΤΟΧΟ

Στο στάδιο αυτό για κάθε individual στον πληθυσμό θα πρέπει να γίνει υπολογισμός της **δυναμικής κατανάλωσης ισχύος**.

Μπορεί να γίνει με υπολογισμό μέσω προσομοίωσης του switching activity κάθε individual.



π.χ. αρχικός πληθυσμός για $L=3$, $N=4$

Individual workload 1

t	a	b	c
t1	1	0	1
t2	0	0	1
t3	1	1	0

Individual workload 2

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Individual workload 3

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

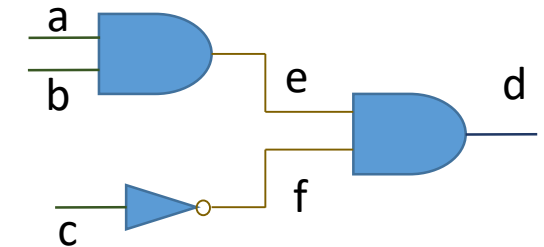
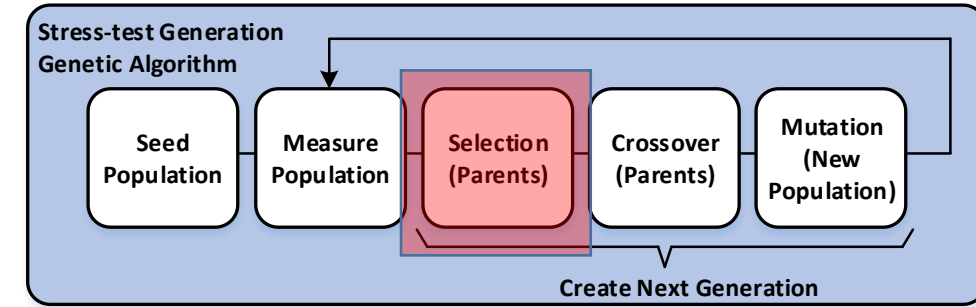
Individual workload 4

t	a	b	c
t1	1	1	1
t2	0	0	0
t3	1	0	1

Βήμα 3^ο: Φυσική επιλογή

Στο στάδιο αυτό επιλέγουμε από τον πληθυσμό ως **γονείς** τα δύο individuals που πετυχαίνουν το καλύτερο αποτέλεσμα στον στόχο που έχουμε θέσει.

Στην προκειμένη περίπτωση επιλέγουμε τα 2 individual workloads με το μεγαλύτερο switching activity και τα ονομάζουμε **γονείς (parents)**.



Individual workload 1

t	a	b	c
t1	1	0	1
t2	0	0	1
t3	1	1	0

Ας υποθέσουμε ότι αυτοί είναι οι 2 γονείς που επιλέγουμε

Individual workload 2

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Individual workload 3

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

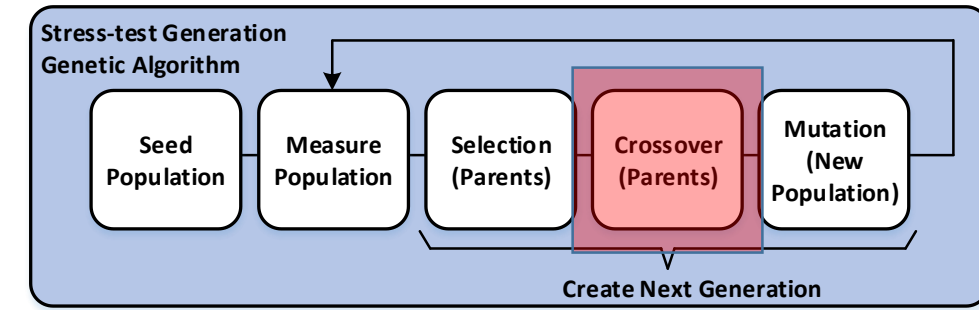
Individual workload 4

t	a	b	c
t1	1	1	1
t2	0	0	0
t3	1	0	1

Βήμα 4^ο: Διασταύρωση

Από τους 2 γονείς παράγουμε έναν νέο πληθυσμό.
Σίγουρα στο νέο πληθυσμό κρατάμε τους γονείς γιατί

- ως την ώρα δίνουν το καλύτερο αποτέλεσμα
- δεν είμαστε ακόμα βέβαιοι ότι τα παιδιά τους θα έχουν καλύτερο αποτέλεσμα



Ας υποθέσουμε ότι αυτοί είναι οι 2 γονείς που επιλέγουμε

Γονέας A

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Γονέας B

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

Individual workload 1

2nd generation

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Individual workload 2

2nd generation

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

Τα υπόλοιπα «παιδιά» παράγονται με διασταύρωση. Μπορεί να γίνει με διάφορους τρόπους. Εμείς θα χρησιμοποιήσουμε τη **τυχαία πρόσμιξη με διαίρεση**. Για κάθε individual/παιδί που θα παράγουμε, θα επιλέγουμε μια **τυχαία γραμμή διαχωρισμού R** από 1 έως L. Οι πρώτες R γραμμές θα επιλέγονται από τον έναν γονέα και οι υπόλοιπες L-R θα επιλέγονται από τον άλλον. Επίσης μπορούμε να αλλάζουμε τυχαία από ποιον γονέα θα επιλέγουμε τις αρχικές γραμμές ρίχνοντας ένα νόμισμα C.

Παράμετροι R,C του αλγορίθμου

C=1, R=2

(όταν C=1, τότε 1^{ος} γονέας ο A)

Individual workload 3

2nd generation

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	0	0	1

C=2, R=1

(όταν C=2, τότε 1^{ος} γονέας ο B)

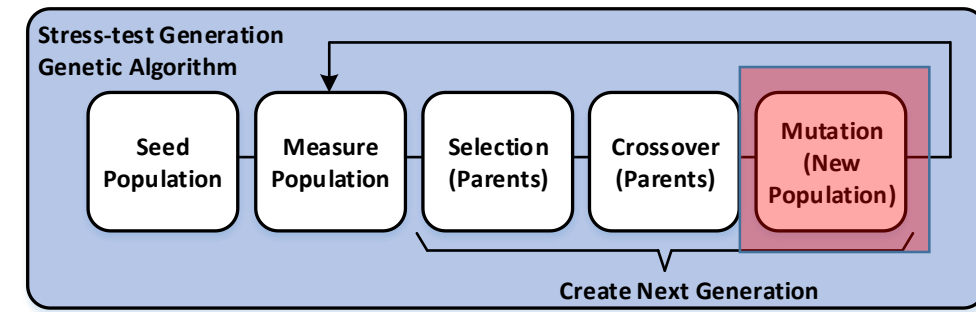
Individual workload 4

2nd generation

t	a	b	c
t1	0	1	1
t2	0	1	0
t3	1	0	1

Βήμα 5^ο: Μετάλλαξη

Με μια πολύ μικρή πιθανότητα που ονομάζεται **συντελεστής μετάλλαξης** (**mutation rate**) **m** αλλάζουμε τα bits των παιδιών που προήλθαν από διαίρεση στο προηγούμενο βήμα. Δεν μεταλλάσσουμε τους δύο γονείς της γενιάς. Το m μπορεί να είναι πολύ μικρό π.χ. $m=0.01$



Individual workload 1
2nd generation

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Individual workload 2
2nd generation

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

Δεν μεταλλάσσονται

C=1, R=2

Individual workload 3
2nd generation

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	0	0	1

C=2, R=1

Individual workload 4
2nd generation

t	a	b	c
t1	0	1	1
t2	0	1	0
t3	1	0	1

Μεταλλάσσονται

Individual workload 1
2nd generation

t	a	b	c
t1	0	0	1
t2	0	1	0
t3	1	0	1

Individual workload 2
2nd generation

t	a	b	c
t1	0	1	1
t2	1	1	1
t3	0	0	1

Mutated individual 3
2nd generation

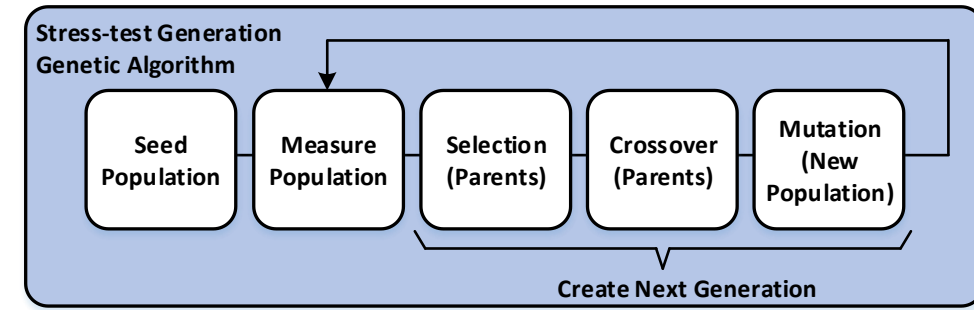
t	a	b	c
t1	0	0	1
t2	0	0	0
t3	0	0	1

Mutated individual 4
2nd generation

t	a	b	c
t1	0	1	0
t2	0	1	0
t3	1	1	1

Παράμετρος του αλγορίθμου m

Σύνοψη παραμέτρων



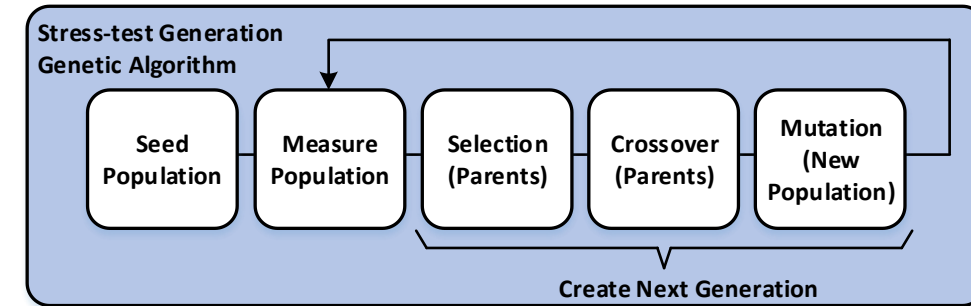
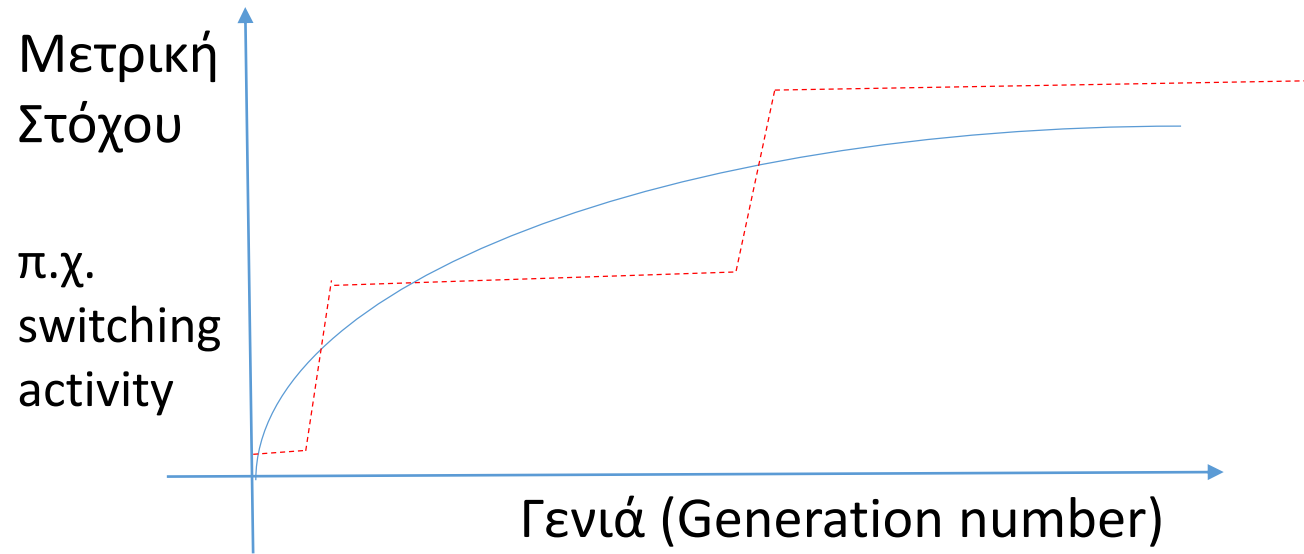
Παράμετροι του αλγορίθμου

- **Μήκος φόρτου εργασίας L (workload length):** είναι το μέγεθος μιας χρονοσειράς και το μετράμε σε πλήθος διανυσμάτων εισόδου.
- **Μέγεθος πληθυσμού N (population size):** το πλήθος των ξεχωριστών φόρτων εργασίας που θα εξερευνάει ο αλγόριθμος σε κάθε του βήμα.
- **Ρυθμός μεταλλάξεων m (mutation rate):** είναι μια πιθανότητα κάποια bits από κάποια παιδιά να αλλάξουν.

Τυχαίες μεταβλητές

- **Τυχαία γραμμή διαχωρισμού R διασταυρώσεων:** είναι το πλήθος των γραμμών που θα επιλεγούν από τον πρώτο γονέα κατά τη διαδικασία της κατασκευής νέων **παιδιών**.
- **Τυχαία νόμισμα C επιλογής πρώτου γονέα:** από τον γονέα αυτών θα επιλέγονται οι πρώτες γραμμές κάποιου παιδιού. Σε κάθε παιδί αλλάζει ο πρώτος τυχαία ο πρώτος γονέας.

Συνεχίζουμε έως ότου δεν υπάρχει βελτίωση



- Στον αλγόριθμο που δείξαμε μπορούμε να βάλουμε διάφορες μετρικές στόχους (objective functions) στο βήμα measure για να βελτιστοποιήσει.
- Δουλεύει όχι μόνο για λογικά κυκλώματα αλλά και για προγράμματα μικροεπεξεργαστών, δηλαδή εντολές assembly
 - Μπορείτε να σκεφτείτε εφαρμογές με άλλες objective functions σε επεξεργαστές;

Εφαρμογές των stress-tests

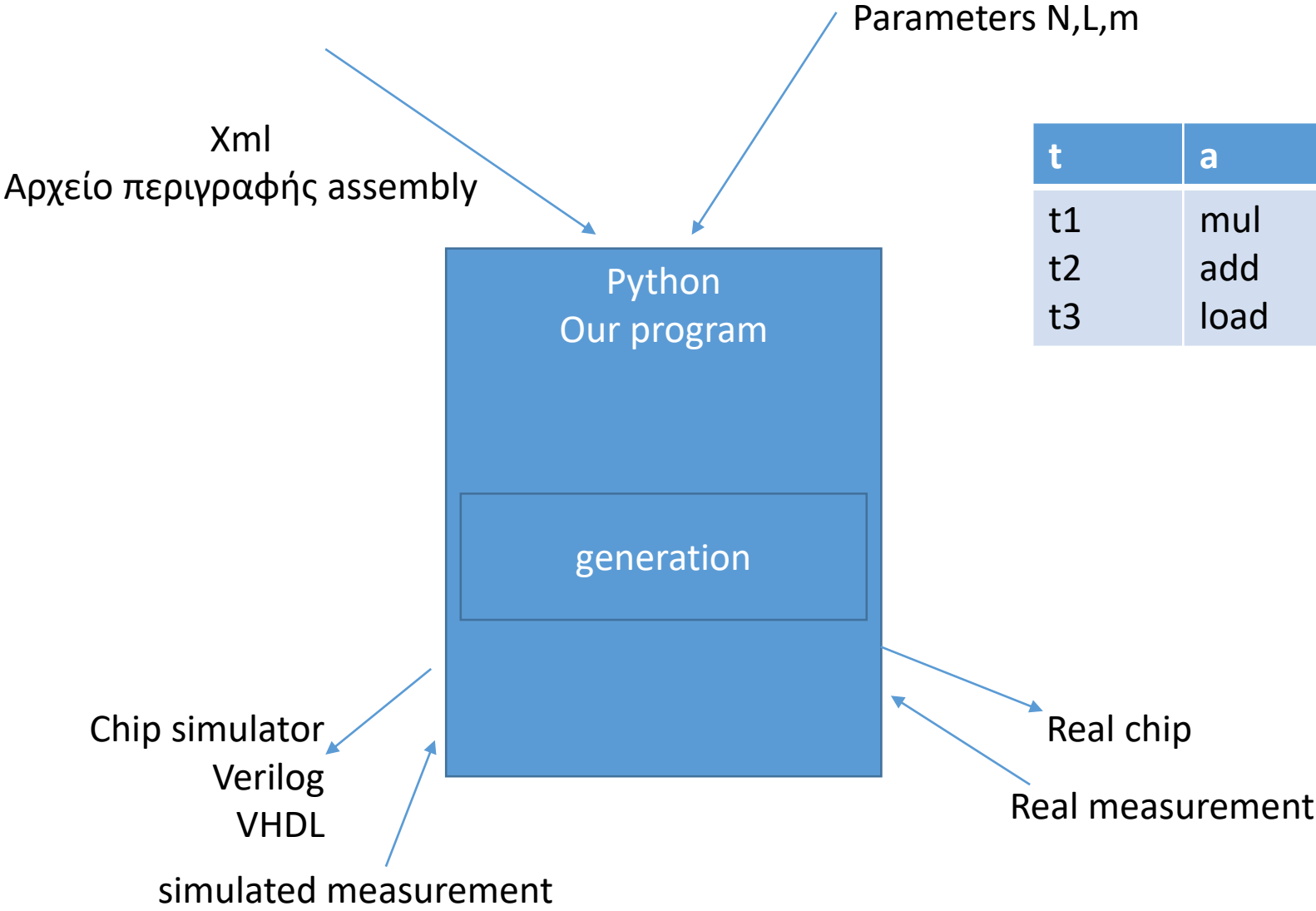
Κατά την κατασκευή των πρωτοτύπων επεξεργαστών, stress-tests που μεγιστοποιούν το **θόρυβο** πάνω σε κυκλώματα, χρησιμοποιούνται για τον υπολογισμό της **ελάχιστης τάσης τροφοδοσίας**.

Stress-tests που μεγιστοποιούν τη **θερμοκρασία** που εκλύεται από τον επεξεργαστή χρησιμοποιούνται για τον υπολογισμό του **συστήματος ψύξης**.

Stress-tests που μεγιστοποιούν την **κατανάλωση ενέργειας/ισχύος** του επεξεργαστή χρησιμοποιούνται για τον σχεδιασμό του **συστήματος τροφοδοσίας**.

Έχουν εφαρμογές επίσης στον χώρο της **ασφάλειας**.

Assembly Stress-tests



t	a	b	c
t1	mul	r1	r2
t2	add	r3	#0003
t3	load	r4	#042F