Switching activity computation - Μοντέλο προσομοίωσης λογικών κυκλωμάτων

Vasileios Tenentes

University of Ioannina

Outline

- Dynamic power and switching activity
- Switching activity of known workload
- Monte Carlo for Switching activity computation, when workload is unknown
- Signal Probabilities for switching activity computation, when workload is unknown
- Εισαγωγή στα μοντέλα προσομοίωσης ψηφιακών κυκλωμάτων

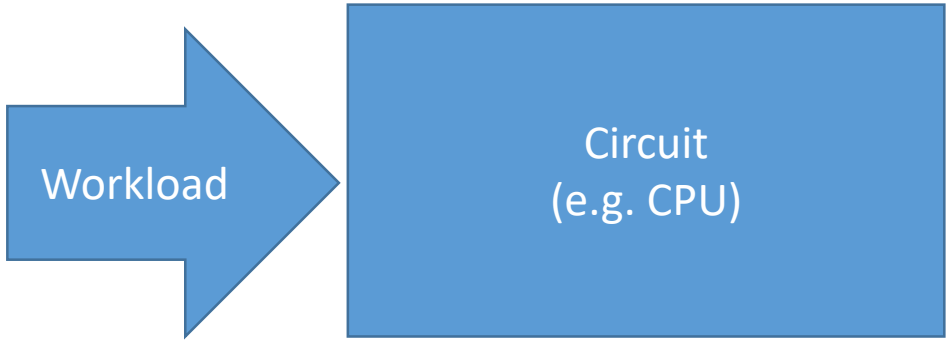
Dynamic Power and Switching activity

- Dynamic power of a circuit depends on its switching activity, which is how often the circuit elements are switching
- Switching activity Esw is the probability of elements (gate, transistors etc.) to switch from 0 to 1 or from 1 to 0:

$$P_d = V^2_{dd} * F_{clk} * C_L * E_{SW} * 0.5$$

F_{clk} = clock frequency, C_L = capacitance seen by gate,
E_{SW} = gate switching activity (0-1 toggles)

Βασικοί όροι



Στο διάγραμμα αυτό
Κύκλωμα **(Circuit):** είναι ένα λογικό κύκλωμα (π.χ. μια CPU ή μια αριθμητική λογική μονάδα) που εκτελεί κάποιους υπολογισμούς

φόρτος εργασίας κυκλώματος (**Workload**): είναι οι χρονοσειρές στις εισόδους του κυκλώματος κατά τη διάρκεια εκτέλεσης ενός προγράμματος

Μπορούμε να βρούμε το switching activity ενός κυκλώματος
Α) με μεγάλη ακρίβεια αν γνωρίζουμε τον φόρτο εργασίας του
Β) κατά προσέγγιση (χρησιμοποιώντας Monte Carlo ή Signal probabilities) όταν δεν γνωρίζουμε το φόρτο εργασίας του

Switching activity of known workload

When the workload (the input vectors) that is executed on the elements is known then the switching activity is computed by **simulating the execution** and counting switches at the output

Example on a 2-input AND gate:

Time	: t1	t2	t3	t4	t5
Input1	: 0	1	0	1	1
Input2	: 1	1	0	0	1
Output	: 0	1	0	0	1

The data above is called **timeseries** or **digital waveforms**.

Beside the timeseries of the workload, **the maximum number of possible switches** at the output of the elements is also needed to compute switching activity. In the case at hand it is: 0->1->0->1->0, which is 4 switches.

Generally for timeseries with N number of vectors, the worst case is N-1.

In this example, we observe 3 number of swithes at the output of the gate. So switching activity is:

Esw=(# of switches observed)/(# of maximum possible switches)=> Esw = 3/4

Switching activity of elements when the workload is not known

When **workload is unknown** then switching activity needs to be statistically evaluated. The task is to identify the most probable switching activity.

This is achieved by either:

1. **Using Monte Carlo**: generating **statistical workload** and transform the problem as a problem of known workload. To transform the problem using Monte Carlo:
   - Consider the inputs of the elements as random variables and take random samples for all of them. These samples, **which are called Monte Carlo permutations,** can be handled as **known workload**.
   - Then perform the switching activity computation of the known statistically generated workload by simulating and counting switches (also consider example MCAND4.m)
2. **Using signal probabilities**: signal probabilities propagation can be used for identifying the switching activity of elements and is presented in the next slide

Monte Carlo for π computation

Η τυχαία μεταβλητή είναι σημεία Si=(xi,yi)

π.χ. S1(x1=0.1,y1=0.9)

Θα χρειαστεί να ελέγξετε αν τα τυχαία επιλεγμένα σημεία είναι εντός ή εκτός του κύκλου (μπορείτε να το κάνετε με την απόσταση του σημείου από το κέντρο του κύκλου)

Έπειτα κρατήστε έναν Counter ελέγχου των σημείων αν είναι εντός ή εκτός του κύκλου

Λόγος= Lim+i_{άπειρο}[[Το πλήθος των σημείων μέσα στον κύκλο]/[Το πλήθος των συνολικών σημείων]=

[εμβαδό κύκλου]/[Εμβαδόν τετραγώνου]

Εμβαδόν κύκλου=πr^2

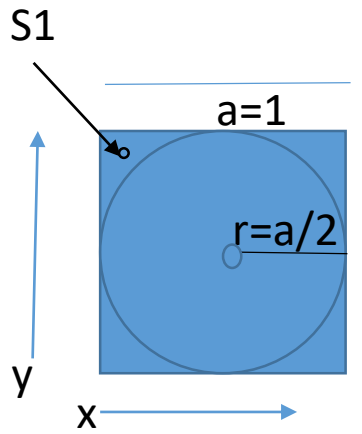
Εμβαδόν τετραγώνου=α^2

r=a/2

[Εμβαδόν κύκλου]/[Εμβαδόν τετραγώνου]= Λόγος ==>

(π(α/2)^2)/(α^2)=Λόγος=π/4=>π=λόγος*4

Όσο πιο πολλά σημεία παίρνετε το αποτέλεσμά σας θα προσεγγίζει το π



PVT Variation

mean

Standard deviation of random variation

Transistor1 random(Vt1,μ=0.5,s=0.02) g s d

Process-Voltage-Temperature variation

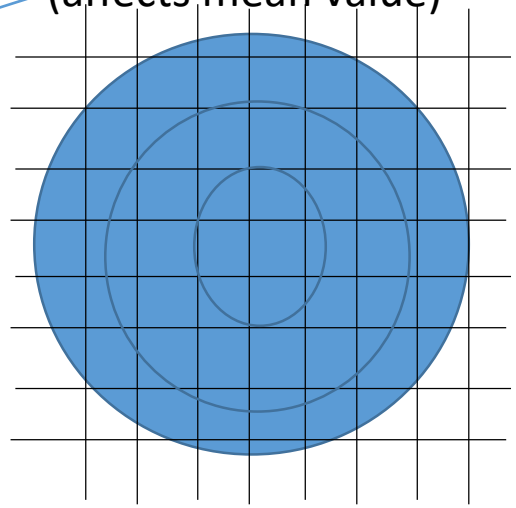
PVT variation

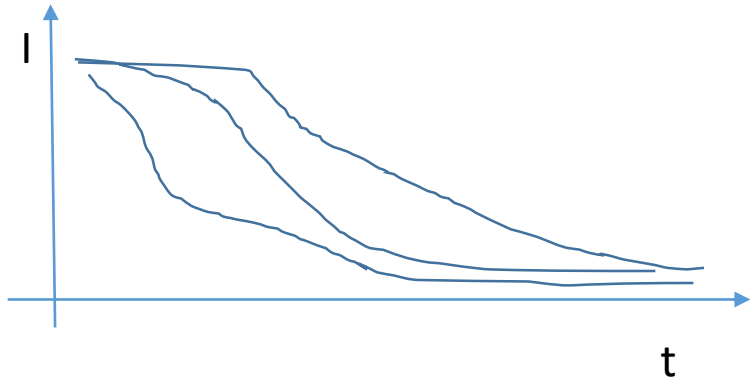
0.7ps 1ps 1.3ps

Vt(bin1)=0.4 +-0.02 V

Vt(bin2)=0.5 +-0.017 V

Systematic variation (affects mean value)





Switching activity of elements when the workload is not known using signal probabilities

Signal probability P of a digital signal is the probability of the signal to be at logic-high (logic-1, ‘1’, ‘true’)

Signal probability propagation process on a logic gate:

1. Signal probability values are set at the inputs of the gate.
2. Signal probability is propagated at the output of the gate using the signal probability function of the gate and the signal probabilities of the inputs.

How to evaluate the **signal probability function** of a gate?

Signal probability function is a function Pout(Pin1, Pin2, …., PinN) of the input signal probabilities. You do not need to remember the signal probability functions of all gates. They can be easily computed using probabilities theory and by examining the truth table of a gate.

To find the function you need to **gather Sum of Products (SOP – άθροισμα γινομένων)** from the truth table. Then you can perform the following check to be sure that the function is correct:

Check: set 0.5 as the signal probability to all inputs (Pin1=Pin2=….=PinN=0.5) and compute the signal probability value at the output. This is performed by counting on the truth table, the number of input combinations that lead to logic-1 at the output. Then, this value must be divided by the total number of input combinations (rows) in the truth table to get the Pout. So when Pin1=Pin2=0.5, Pout must honour:
Check=(# of rows in the truth table with Out=1)/(# total number of rows in the truth table).

To finalise the check, replace Pin1,Pin2,….,PinN with **0.5** value at the computed signal probability function. It must be equal to **Check.**

Examples in the next slide

Switching activity of elements when the workload is not known using signal probabilities

Truth table 2-input AND

In1	In2	Out
0	0	0
0	1	0
1	0	0
1	1	1

example on a 2-input AND gate

- **Sum of Products:** from the truth table of the 2-input AND gate signal Out is 1 only when In1=In2=1.
  Therefore, the candidate function as a sum of product of this, it becomes: **Pout_AND(Pin1, Pin2)=Pin1*Pin2**

- **Check**: The output becomes logic-`1' only during one input combination (out of the 4 input combinations). So it is **Check=1/4**. The check is successful because Pout_AND(0.5, 0.5)=0.5*0.5=0.25=1/4

Truth table 2-input OR

In1	In2	Out
0	0	0
0	1	1
1	0	1
1	1	1

example on a 2-input OR gate

- **Sum of Products:** from the truth table of the 2-input OR gate, signal Out is 1 in three cases: when In1=In2=1; when In1=1 and In2=0; and when In1=0 and In2=1. So the candidate function as a sum of products of these, it becomes: **Pout_OR(Pin1, Pin2)= Pin1*Pin2 + Pin1*(1-Pin2) + (1-Pin1)*Pin2**

- **Check**: The output becomes logic-`1' during three input combination (out of the 4 input combinations). So it is **Check=3/4**. The check is successful because:
  Pout_OR(0.5, 0.5)=0.5*0.5+0.5*(1-0.5)+(1-0.5)*0.5=> Pout_OR(0.5,0.5)=0.75=3/4

Switching Activity Computation

The switching activity of a gate is then computed using its output signal probability: a gate's output switches when two successive values of it are complementary. This occurs either: (1) when a bit is '1' and the successive bit is '0'; (2) or when a bit is '0' and the successive bit is '1':

So it is Esw=Pout*(1-Pout)+(1-Pout)*Pout=> **Esw=2*Pout(Pin1, Pin2)(1-Pout(Pin1, Pin2)).** With this equation, switching activity can be computed, when signal probabilities at the inputs is known. Note that if we consider the probability at the inputs to be 0.5, then this formula collapses to **Esw=2*Pout(0.5,0.5)(1-Pout(0.5,0.5))➔ Esw=2*Check*(1-Check)**.

Για σήμα με γνωστό signal probability p ισχύει ότι: Esw=2*p*(1-p)

Some tricks:

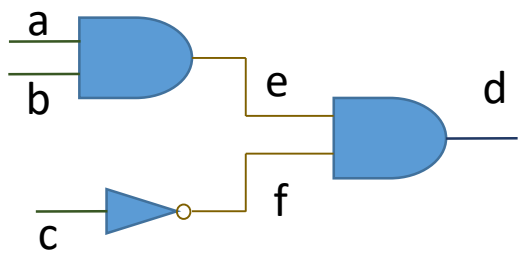
- Signal probabilities of complementary gates are also complementary. Example: Pout_OR=1-Pout_NOR

- Although the check must be honoured you can use the Monte Carlo technique to evaluate if the probability function considered for a gate is correct or not.

Όμως τι γίνεται με κυκλώματα πιο πολύπλοκα από μια λογική πύλη;

- Για να βρούμε το switching activity ενός κυκλώματος πρέπει να βρούμε το switching activity όλων των δομικών του **στοιχείων**. Τα **δομικά στοιχεία** μπορεί να είναι πιο πολύπλοκα από λογικές πύλες αλλά στην περίπτωσή μας θα μας απασχολήσουν μόνο οι λογικές πύλες

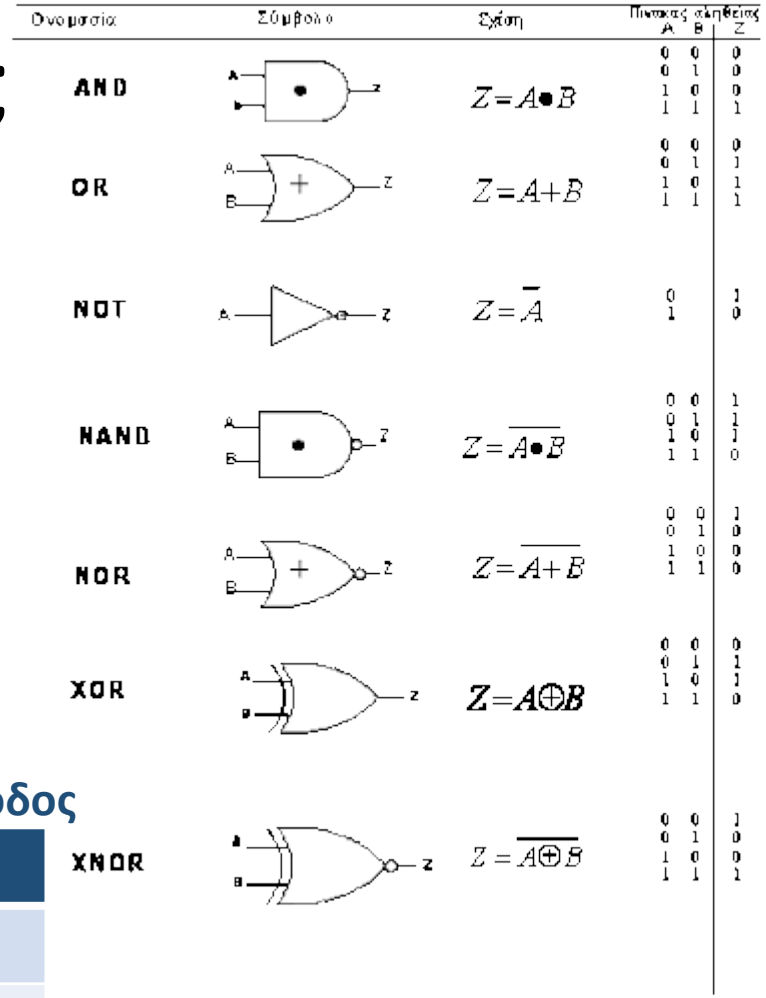
Παράδειγμα



Με γνωστό φόρτο εργασίας

	φόρτος εργασίας			Ενδιάμεσες έξοδοι		Τελική έξοδος
χρόνος	a	b	c	e	f	d
t1	0	1	0			?
t2	1	1	0			?
t3	1	0	0			?
t4	0	0	1			?

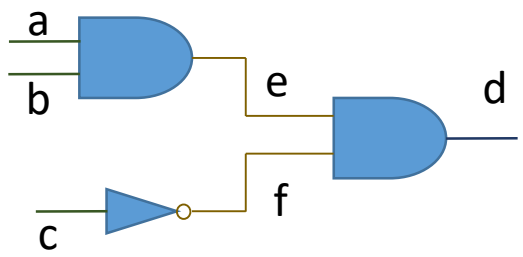
Βασικές Λογικές Πύλες



Όμως τι γίνεται με κυκλώματα πιο πολύπλοκα από μια λογική πύλη;

- Για να βρούμε το switching activity ενός κυκλώματος πρέπει να βρούμε το switching activity όλων των δομικών του **στοιχείων**. Τα **δομικά στοιχεία** μπορεί να είναι πιο πολύπλοκα από λογικές πύλες αλλά στην περίπτωσή μας θα μας απασχολήσουν μόνο οι λογικές πύλες

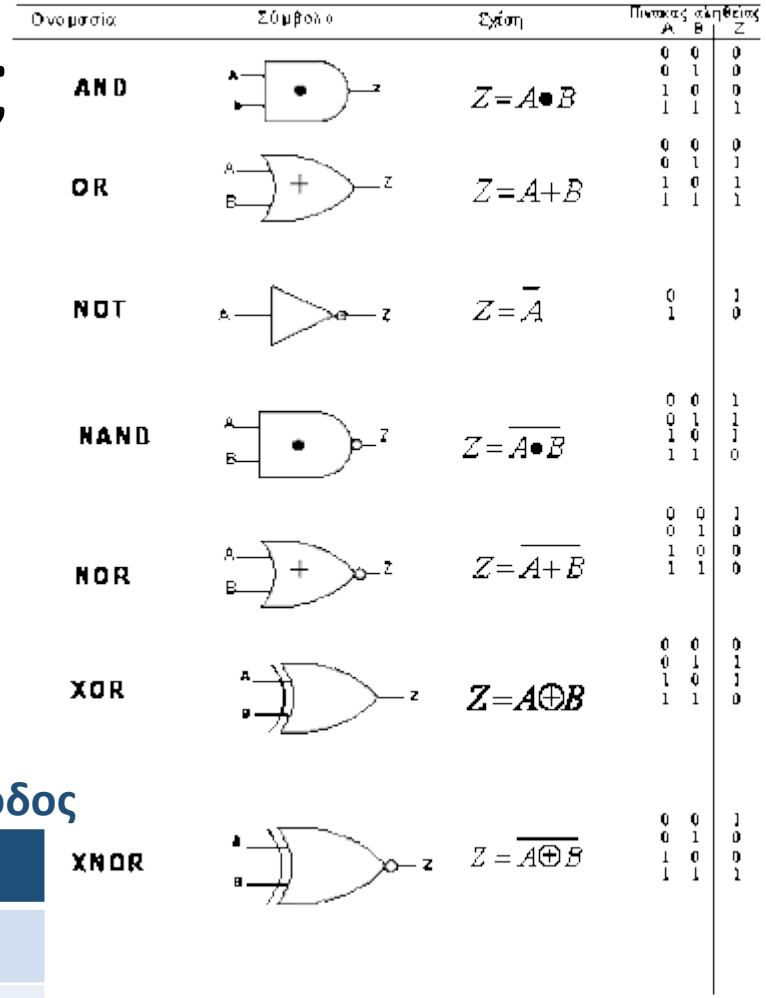
Παράδειγμα



Με γνωστό φόρτο εργασίας

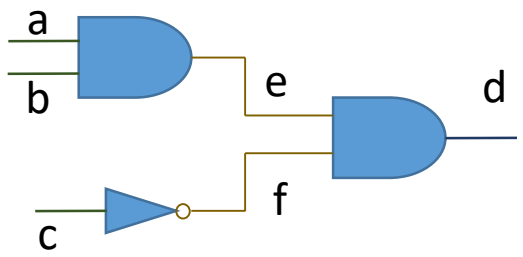
	φόρτος εργασίας			Ενδιάμεσες έξοδοι		Τελική έξοδος
χρόνος	a	b	c	e	f	d
t1	0	1	0	0	1	0
t2	1	1	0	1	1	1
t3	1	0	0	0	1	0
t4	0	0	1	0	0	0

Βασικές Λογικές Πύλες

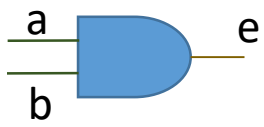


Ανάλυση και των ενδιάμεσων εξόδων

	φόρτος εργασίας			Ενδιάμεσες έξοδοι		Τελική έξοδος
χρόνος	a	b	c	e	f	d
t1	0	1	0	0	1	0
t2	1	1	0	1	1	1
t3	1	0	0	0	1	0
t4	0	0	1	0	0	0



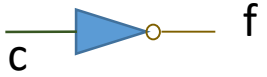
έξοδος e



t	a	b	e
t1	0	1	0
t2	1	1	1
t3	1	0	0
t4	0	0	0

Esw(e)=2/3

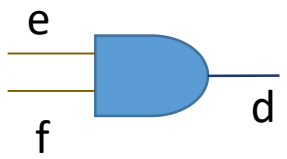
έξοδος f



t	c	f
t1	0	1
t2	0	1
t3	0	1
t4	1	0

Esw(f)=1/3

έξοδος d



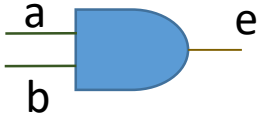
t	e	f	d
t1	0	1	0
t2	1	1	1
t3	0	1	0
t4	0	0	0

Esw(d)=2/3

Θυμηθείτε: **Esw=(# of switches) / (max # of switches)**

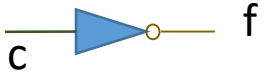
Ανάλυση και των ενδιάμεσων εξόδων

έξοδος e



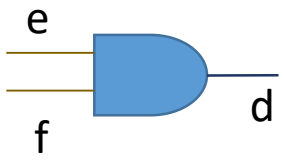
Esw(e)=2/3

έξοδος f



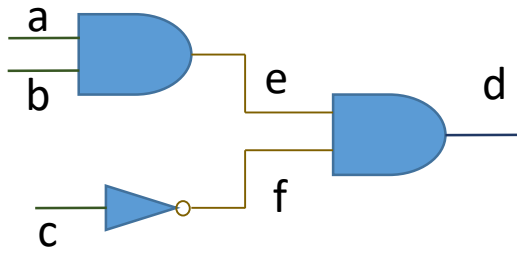
Esw(f)=1/3

έξοδος d



Esw(d)=2/3

circuit



Πρέπει να εφαρμόσουμε τον παρακάτω τύπο σε κάθε στοιχείο ξεχωριστά με βάση το δικό του C_L ώστε να βρούμε τα Pd(e), Pd(f) και Pd(d)

$$P_d = V^2_{dd} * F_{clk} * C_L * E_{SW} * 0.5$$

F_{clk} = clock frequency, C_L = capacitance seen by gate,
E_{SW} = gate switching activity (0-1 toggles)

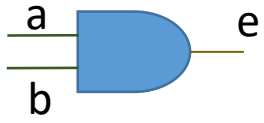
Και στο τέλος να βρούμε την τιμή:
Pdynamic(circuit)= Pd(e)+Pd(f) +Pd(d)

Σύνοψη: Για να βρούμε το switching activity ενός κυκλώματος με γνωστό φόρτο εργασίας πρέπει:

1. Να προσομοιώσουμε το κύκλωμα για το φόρτο εργασίας
2. Για κάθε έξοδο κάθε στοιχείου του κυκλώματος να υπολογίσουμε το switching activity

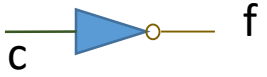
Ανάλυση και των ενδιάμεσων εξόδων

έξοδος e



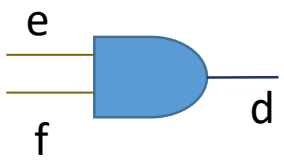
Esw(e)=2/3

έξοδος f



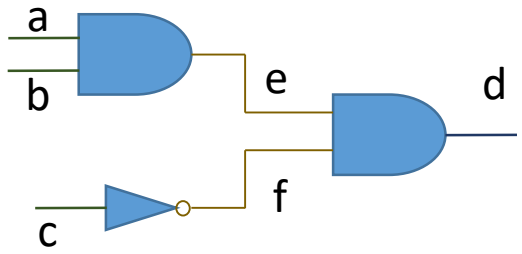
Esw(f)=1/3

έξοδος d



Esw(d)=2/3

circuit



Πρέπει να εφαρμόσουμε τον παρακάτω τύπο σε κάθε στοιχείο ξεχωριστά με βάση το δικό του C_L ώστε να βρούμε τα Pd(e), Pd(f) και Pd(d)

$$P_d = V^2_{dd} * F_{clk} * C_L * E_{SW} * 0.5$$

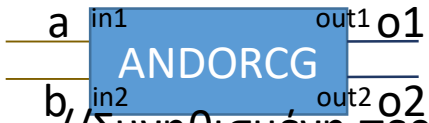
F_{clk} = clock frequency, C_L = capacitance seen by gate,
E_{SW} = gate switching activity (0-1 toggles)

Και στο τέλος να βρούμε την τιμή:
Pdynamic(circuit)= Pd(e)+Pd(f) +Pd(d)

```
//Αρχείο περιγραφής κυκλώματος
e =AND(a,b)
f =NOT(c)
d =AND(e,f)
```

```
//Συνηθίζεται αυτή η μορφή αρχείου
//περιγραφής κυκλώματος
AND(e, a, b)
NOT(f, c)
AND(d, e, f)
```

ANDOR(o1,o2, a, b)




```
//Συνηθισμένη περιγραφή κυκλώματος που δεν παίζει ρόλο η σειρά
ANDOR(a(in1), b(in2), out1(o1),out2(o2))
```