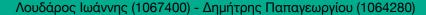
# Σχεδιασμός Συστημάτων VLSI

### Εργαστήριο 1





Μπορείτε να δείτε την τελευταία έκδοση του Project εδώ ή σκανάροντας τον κωδικό QR που βρίσκεται στην επικεφαλίδα.

### Περιγραφή Αναφοράς

Παρακάτω παραθέτω τις απαντήσεις μας στην "Πρώτη Εργαστηριακή Άσκηση" του μαθήματος "Σχεδιασμός Συστημάτων VLSI" καθώς και σχόλια τα οποία προέκυψαν κατά την εκπόνηση του.

## Περιεχόμενα

1.	Ζητούμενο 1	2
	΄ ΄ Ζητούμενο 2	
	Ζητούμενο 3	

## Απαντήσεις

#### 1. Ζητούμενο 1.

Τροποποιήσαμε το παράδειγμα 3. Ο κώδικας που χρησιμοποιήσαμε φαίνεται παρακάτω:

```
initial begin
    a=0;
    b=0;
    repeat(256) begin
        repeat(256) begin
        #(delay) sum = a+b;
        if (out == sum) check=1'b1;
        else check=1'bX;
        b=b+1;
    end
    a=a+1;
end
```

Αν θέλουμε να το κάνουμε με for, χρειάζεται είτε να αυξήσουμε τον καταχωρητή α, είτε να χρησιμοποιήσουμε το break υπό συνθήκη ώστε να καταφέρουμε να εξετάσουμε όλες τις τιμές.

#### 2. Ζητούμενο 2.

Δημιουργήσαμε ένα παράδειγμα που βάζει στο mac 10 τυχαίες τιμές και καταγράφει εισόδους και αποτελέσματα στο αρχείο "mac\_tb\_result.txt". Ύστερα ένα python script ελέγχει το txt. Για την εκτέλεση του προσθέσαμε στο Makefile το zitoumeno2.

```
zitoumeno2: mac.o mac_tb.o
    $(VCS) $(VCS_OPTS) $(SRC_DIR)/mac_tb.v $(SRC_DIR)/mac.v -o zitoumeno2
    python3 $(SRC_DIR)/test_mac_results.py mac_tb_results.txt
```

Στα αποτελέσματα που έχουμε ανεβασμένα, το τεστ αποτυγχάνει! Αυτό συμβαίνει γιατί ο καταχωρητής φτάνει σε overflow.

```
module mac_tb;
   parameter input_width = 4;
   parameter sum_width = 10;
   parameter fileout = "mac_tb_result.txt";
     reg [input_width-1:0] a,b;
wire [sum_width-1:0] sum;
     reg clock, reset;
integer f;
      mac #(.iwidth(input_width), .swidth(sum_width)) testmac (
            .a_i(a),
.b_i(b),
            .sum_o(sum),
            .clk(clock),
.rstn(reset)
      );
            clock = 0;
            reset = 1;
            #2 reset = 0;
            #2 reset = 1;
      always #5 clock = ~clock;
            f = $fopen(fileout, "w");
repeat(20) begin
                  a = $random;
b = $random;
                  #10;
            $finish;
            repeat(20) begin
    $fwrite(f,"%d %d %d\n", a, b, sum );
                  #10;
endmodule
```

```
import sys
a=[]
b=[]
out=[]
out=[]
sum=0

with open(sys.argv[1]) as f:
    for line in f:
        nums = [int(n) for n in line.split()]
        a.append(nums[0])
        b.append(nums[1])
        out.append(nums[2])

for i in range(len(out)):
        sum = sum + a[i] * b[i]

        if sum ==out[i]:
            print("Line ",i+1," ok!")
        else:
            print("[!] Line ",i+1," ERROR")
            error=1

if error==1:
    print("The test failed!")
else:
    print("The test was succesful!")
```

test\_mac\_results.py

#### 3. Ζητούμενο 3.

```
module count #(
    parameter c0_size = 3,
    parameter c1_size = 16
)(
    input clock, reset, enable,
    output reg [c0_size-1:0] c0,
    output reg [c1_size-1:0] c1
);

always @(posedge clock or negedge reset) begin
    if (!reset) begin
        c0 = 0;
        c1 = 2;
    end
    else if (enable)
    begin
        c0 = c0 + 1;
        if (c0==0)
              c1 = c1+1;
    end
end
end
end
```

count\_tb.v