Homework 1

黃偉祥 X1136010

How to

- In PUF/ and Bonus/
- `make sim file=testbench_file.v`
- Data collected on `140.115.71.44` server, using `my_ta_tb.v`, `my_tb.v`, and `my_UQ_tb.v` to simulate and output as `my_ta_tb.fsdb`, `my__tb.fsdb`, and `UQ_tb.fsdb`
- `make view file=target_file.fsdb`
- UF calculation by when `ready == 1`, check the response
- UQ calculation by the program, when all_ready have collected, then display the response and calculate the UQ by the formula

```
// To get which PUF is ready
always @(posedge ready[0])begin
    if(ready[0]) begin
        all_ready[0] = 1;
        all_response[0]=response[0];
    end
end

// To calculate UQ and show response
always @(*) begin
        if (all_ready == 10'b1111111111) begin // If all PUF is ready
            $display("CHIP 0 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 1 : at time %t, chall_in = %h, response = %h, ready = %b", $time
            $display("CHIP 2 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 3 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 4 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 5 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 6 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 7 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 8 : at time %t, chall_in = %h, response = %h, ready = %b", $tim
            $display("CHIP 9 : at time %t, chall_in = %h, response = %h, ready = %b", $tim

            // Calculate UQ for all PUF
            UQ = 0;
```

```
        total_hd=0;
        for (i=0;i<10;i=i+1)begin
            for(j=i+1;j<10;j=j+1)begin
                hd = 0;
                for (b = 0; b<9;b=b+1)begin
                    if(all_response[i][b] != all_response[j][b])begin
                        hd = hd+1;
                    end
                end
                total_hd = total_hd + hd;
            end
        end
        UQ = (2.0/(10.0*9.0))*(total_hd / 8.0);
        $display("UQ = %0.2f",UQ);
        all_ready<=0;
    end
end
```

Design Philosophy

- Scrambler
  - Create a state to store the input for LSFR feedback, a feedback new_bit for feedback loop
  - new_bit use XOR(^) from state to generate another new bit for state in next cycle
  - check `if rst ? state = in : shift state`
  - output = state XOR in
- MUX
  - `output = in[sel]`, sel is 0~15 and it is to select which input should go to output, so just select in[sel] to output
- Counter
  - create a reg count_reg for count to maximum and output 1 when it reach maximum
  - Follow the instruction in template to done everything else
- Race_Arbiter
  - if rst then result is 0 else
    - if a == 1 then result is 1, else if b == 1 then result is 0 else stay
  - if rst then done is 0 else

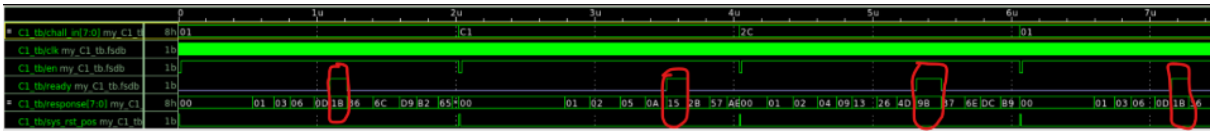
        - if a or b then is 1
- Buffer
    - create a reg bit_count to count if all bit collected
    - if rst then all set to 0 and buf_rst set to 1 (to reset counter and race_arbiter) else
        - if arbiter done then `buf_rst=1`, put arbiter result into response and shift, bit_count ++, if bit_count to max then ready set to 1 else ready set to 0 else
            - `buf_rst = 0`

Simulation waveform

- Using `C1_tb.v`, `C2_tb.v`, and `C3_tb.v` and output as `my_C1_tb.fsdb`, `my_C2_tb.fsdb`, and `my_C3_tb.fsdb`

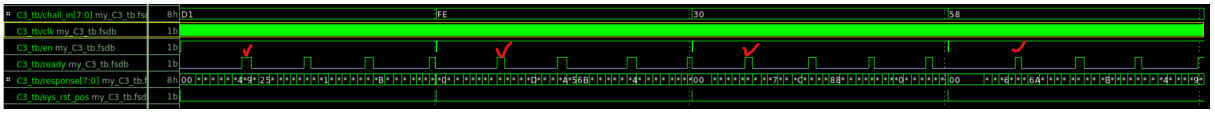
1. Simulated by `C1_tb.v`

- Output is when ready = 1, so first response = 1B (hexadecimal)



2. Simulated by `C2_tb.v`

3. Simulated by `C2_tb.v`



- tick is the first output from different challenges after reset

---

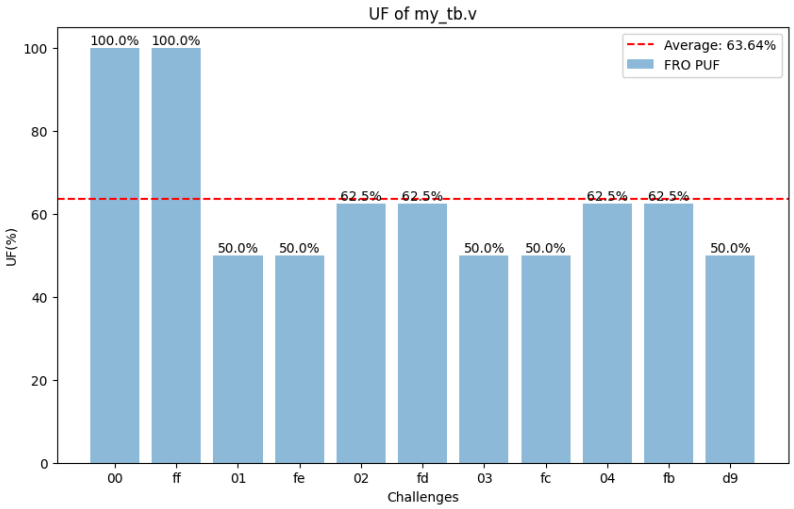
Feedforward Ring Oscillator PUF Result (UF&UQ)

Uniformity(UF)

1. `my_tb.v`

```
1 challenge
At time            1355000, chall_in = 00, response = ff, ready = 1
2 challenge
At time            3375000, chall_in = ff, response = ff, ready = 1
3 challenge
At time            5135000, chall_in = 01, response = 1b, ready = 1
4 challenge
At time            7155000, chall_in = fe, response = 1b, ready = 1
5 challenge
At time            9255000, chall_in = 02, response = 8f, ready = 1
6 challenge
At time           11275000, chall_in = fd, response = 8f, ready = 1
7 challenge
At time           13245000, chall_in = 03, response = 53, ready = 1
8 challenge
At time           15265000, chall_in = fc, response = 53, ready = 1
9 challenge
At time           17365000, chall_in = 04, response = 97, ready = 1
10 challenge
At time           19385000, chall_in = fb, response = 97, ready = 1
11 challenge
At time           21385000, chall_in = d9, response = 1b, ready = 1
12 challenge
At time           23405000, chall_in = d9, response = 1b, ready = 1
```

- Including **00** and **ff** challenge, using **1 FRO PUF** and **12 challenges**
- 8/8 +8/8 + 4/8+4/8 + 5/8+5/8 +4/8 + 4/8 + 5/8 + 5/8 + 4/8 + 4/8
- Average = **64.5%**
- **0xd9** and **0x01** challenge have same response
- `(01,fe)` , `( 02,fd )` , `( 03,fc )` , `( 04,fb )` have same result infer that maybe at most 2^7 results

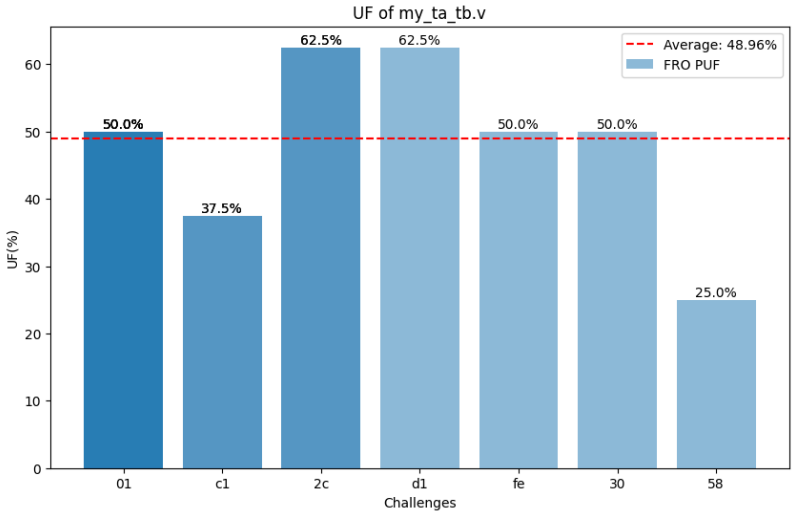


- Same input have same output → Stability

2. `my_ta_tb.v`

```
1 challenge
At time            1095000, chall_in = 01, response = 1b, ready = 1
2 challenge
At time            3515000, chall_in = c1, response = 15, ready = 1
3 challenge
At time            5315000, chall_in = 2c, response = 9b, ready = 1
4 challenge
At time            7155000, chall_in = 01, response = 1b, ready = 1
5 challenge
At time            9175000, chall_in = 01, response = 1b, ready = 1
6 challenge
At time           11595000, chall_in = c1, response = 15, ready = 1
7 challenge
At time           13395000, chall_in = 2c, response = 9b, ready = 1
8 challenge
At time           15235000, chall_in = 01, response = 1b, ready = 1
9 challenge
At time           17325000, chall_in = d1, response = c7, ready = 1
10 challenge
At time           19275000, chall_in = fe, response = 1b, ready = 1
11 challenge
At time           21365000, chall_in = 30, response = 1d, ready = 1
12 challenge
At time           23355000, chall_in = 58, response = 11, ready = 1
```

- Use C1 to C3 as challenges, using **1 FRO PUF** and **12 challenges**
- 50% + 37.5% + 62.5% + 50% + 50% + 37.5% + 62.5% + 50% + 62.5% + 50% + 50% + 25%
- Average : **48.96%**

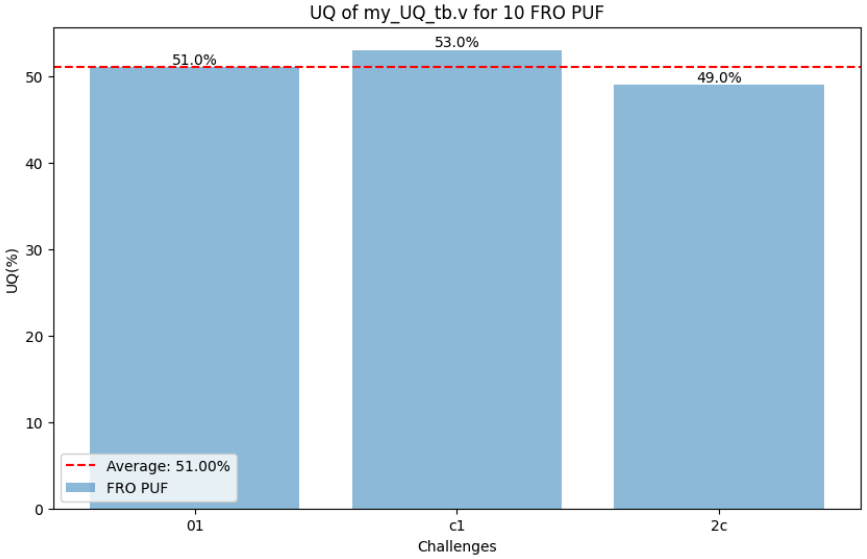


Uniqueness (UQ)

1. my_UQ_tb.v

```
*Verdi* : Create FSDB file 'UQ_tb.fsdb'
*Verdi* : Begin traversing the scope (UQ_tb), layer (0).
*Verdi* : End of traversing.
First challenge
CHIP 0 : at time              1485000, chall_in = 01, response = ae, ready = 1
CHIP 1 : at time              1485000, chall_in = 01, response = 7e, ready = 1
CHIP 2 : at time              1485000, chall_in = 01, response = 73, ready = 1
CHIP 3 : at time              1485000, chall_in = 01, response = db, ready = 1
CHIP 4 : at time              1485000, chall_in = 01, response = 13, ready = 1
CHIP 5 : at time              1485000, chall_in = 01, response = 65, ready = 1
CHIP 6 : at time              1485000, chall_in = 01, response = 34, ready = 1
CHIP 7 : at time              1485000, chall_in = 01, response = 4d, ready = 1
CHIP 8 : at time              1485000, chall_in = 01, response = 12, ready = 1
CHIP 9 : at time              1485000, chall_in = 01, response = 4b, ready = 1
UQ = 0.51
Second challenge
CHIP 0 : at time              3435000, chall_in = c1, response = 55, ready = 1
CHIP 1 : at time              3435000, chall_in = c1, response = df, ready = 1
CHIP 2 : at time              3435000, chall_in = c1, response = 22, ready = 1
CHIP 3 : at time              3435000, chall_in = c1, response = 3e, ready = 1
CHIP 4 : at time              3435000, chall_in = c1, response = ea, ready = 1
CHIP 5 : at time              3435000, chall_in = c1, response = 0a, ready = 1
CHIP 6 : at time              3435000, chall_in = c1, response = 1d, ready = 1
CHIP 7 : at time              3435000, chall_in = c1, response = d9, ready = 1
CHIP 8 : at time              3435000, chall_in = c1, response = c3, ready = 1
CHIP 9 : at time              3435000, chall_in = c1, response = be, ready = 1
UQ = 0.53
Third challenge
CHIP 0 : at time              5455000, chall_in = 2c, response = e1, ready = 1
CHIP 1 : at time              5455000, chall_in = 2c, response = c6, ready = 1
CHIP 2 : at time              5455000, chall_in = 2c, response = bd, ready = 1
CHIP 3 : at time              5455000, chall_in = 2c, response = 5b, ready = 1
CHIP 4 : at time              5455000, chall_in = 2c, response = 03, ready = 1
CHIP 5 : at time              5455000, chall_in = 2c, response = 51, ready = 1
CHIP 6 : at time              5455000, chall_in = 2c, response = ff, ready = 1
CHIP 7 : at time              5455000, chall_in = 2c, response = 41, ready = 1
CHIP 8 : at time              5455000, chall_in = 2c, response = eb, ready = 1
CHIP 9 : at time              5455000, chall_in = 2c, response = 19, ready = 1
UQ = 0.49
Fourth challenge
CHIP 0 : at time              7545000, chall_in = 01, response = ae, ready = 1
CHIP 1 : at time              7545000, chall_in = 01, response = 7e, ready = 1
CHIP 2 : at time              7545000, chall_in = 01, response = 73, ready = 1
CHIP 3 : at time              7545000, chall_in = 01, response = db, ready = 1
CHIP 4 : at time              7545000, chall_in = 01, response = 13, ready = 1
CHIP 5 : at time              7545000, chall_in = 01, response = 65, ready = 1
CHIP 6 : at time              7545000, chall_in = 01, response = 34, ready = 1
CHIP 7 : at time              7545000, chall_in = 01, response = 4d, ready = 1
CHIP 8 : at time              7545000, chall_in = 01, response = 12, ready = 1
CHIP 9 : at time              7545000, chall_in = 01, response = 4b, ready = 1
UQ = 0.51
Simulation complete via $finish(1) at time 8080 NS + 0
./my_UQ_tb.v:462         $finish;
```

- Challenges = **01,c1,2c,01**, using **10** different **FRO PUF** with **3 groups** of **challenges** and **1 group** to check **stability**.
- Average UQ = (51+53+49) / 3 = **51%**

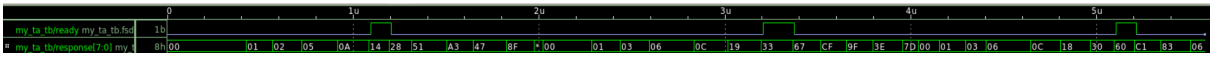


Bonus

Simulation waveform

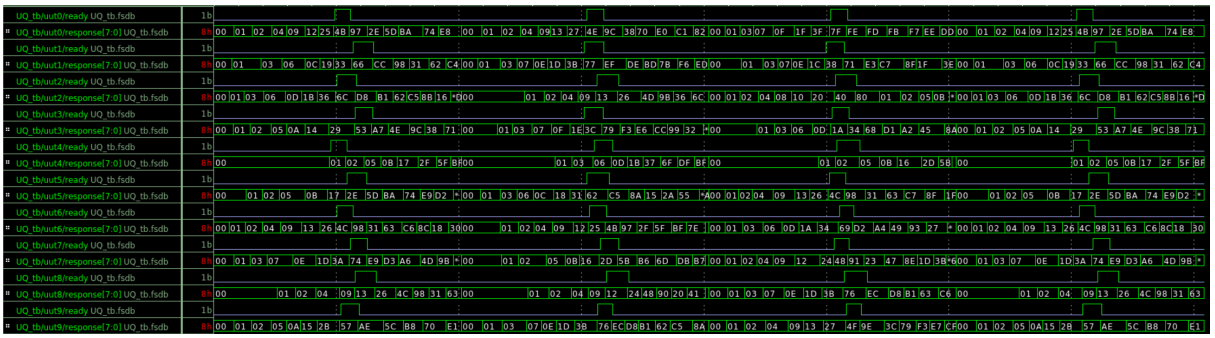
Only show part of challenge of waveform, because is too long

1. Simulated by `my_ta_tb.v`



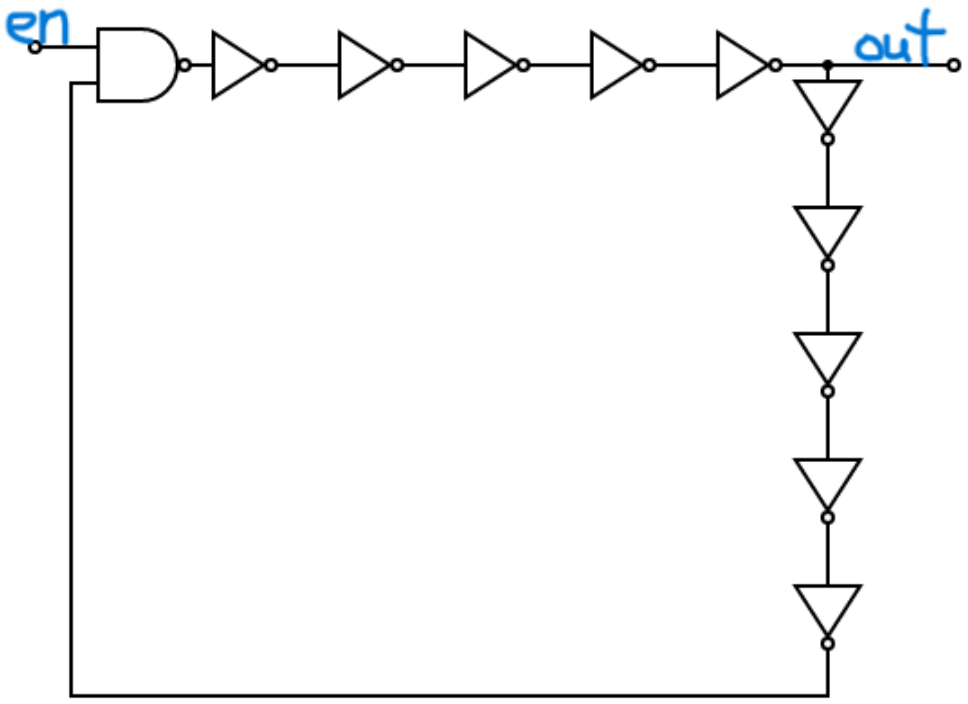
2. Simulated by `my_tb.v`

3. Simulated by `my_UQ_tb.v`



RO PUF Result (UF&UQ)

RO design



Scrambler

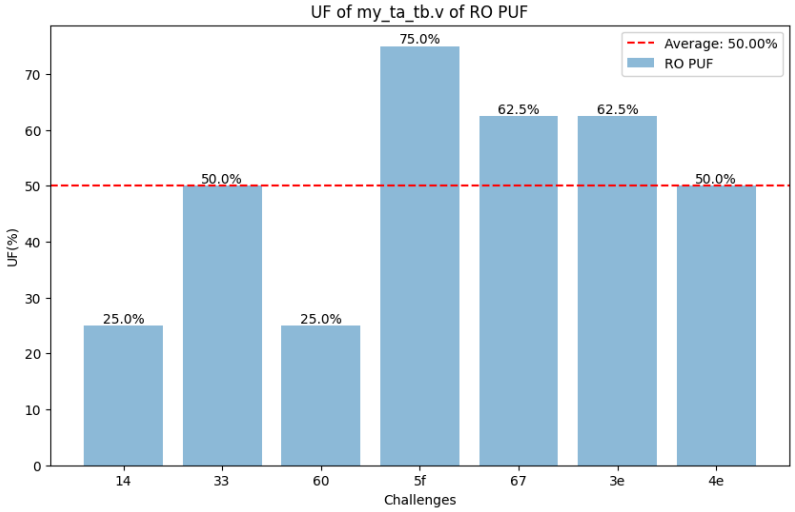
$$m(x) = x^8 + x^4 + x^3 + x + 1$$

Uniformity(UF)

1. my_ta_tb.v

```
*Verdi* : Create FSDB file 'my_ta_tb.fsdb'
*Verdi* : Begin traversing the scope (my_ta_tb), layer (0).
*Verdi* : End of traversing.
1 challenge
At time             1095000, chall_in = 01, response = 14, ready = 1
2 challenge
At time             3205000, chall_in = c1, response = 33, ready = 1
3 challenge
At time             5105000, chall_in = 2c, response = 60, ready = 1
4 challenge
At time             7155000, chall_in = 01, response = 14, ready = 1
5 challenge
At time             9175000, chall_in = 01, response = 14, ready = 1
6 challenge
At time            11285000, chall_in = c1, response = 33, ready = 1
7 challenge
At time            13185000, chall_in = 2c, response = 60, ready = 1
8 challenge
At time            15235000, chall_in = 01, response = 14, ready = 1
9 challenge
At time            17195000, chall_in = d1, response = 5f, ready = 1
10 challenge
At time            19355000, chall_in = fe, response = 67, ready = 1
11 challenge
At time            21235000, chall_in = 30, response = 3e, ready = 1
12 challenge
At time            23375000, chall_in = 58, response = 4e, ready = 1
```

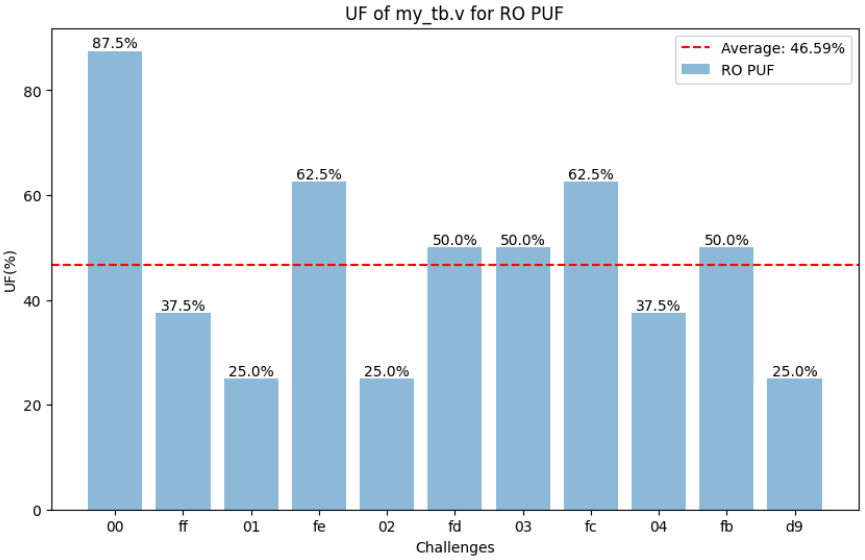
- Using **1 RO PUF** and **12 challenges**
- (0x14)25% + (0x33)50% + (0x60)25% + (0x5f)75% + (0x67)62.5% + (3e)62.5% + (0x4e)50%
- Average = **50%**



2. my_tb.v

```
*Verdi* : Create FSDB file 'my__tb.fsdb'
*Verdi* : Begin traversing the scope (my__tb), layer (0).
*Verdi* : End of traversing.
1 challenge
At time              1195000, chall_in = 00, response = 7f, ready = 1
2 challenge
At time              3025000, chall_in = ff, response = 19, ready = 1
3 challenge
At time              5135000, chall_in = 01, response = 14, ready = 1
4 challenge
At time              7235000, chall_in = fe, response = 67, ready = 1
5 challenge
At time              9145000, chall_in = 02, response = 42, ready = 1
6 challenge
At time             11165000, chall_in = fd, response = 4b, ready = 1
7 challenge
At time             13275000, chall_in = 03, response = 35, ready = 1
8 challenge
At time             15195000, chall_in = fc, response = 37, ready = 1
9 challenge
At time             17225000, chall_in = 04, response = 64, ready = 1
10 challenge
At time             19205000, chall_in = fb, response = 78, ready = 1
11 challenge
At time             21395000, chall_in = d9, response = 0c, ready = 1
12 challenge
At time             23415000, chall_in = d9, response = 0c, ready = 1
```

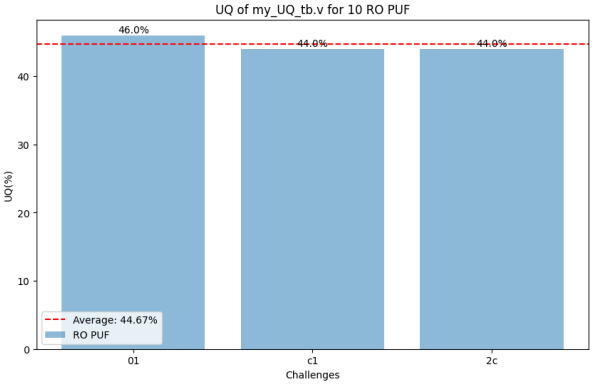
- Using **1 RO PUF** and **12 challenges**
- 87.5% + 37.5% + 25% + 62.5% + 25% + 50% + 50% + 62.5% + 37.5% + 50% + 25%
- Average UF = **46.59%**



3. `my_UQ_tb.v`

```
*Verdi* : Create FSDB file 'UQ_tb.fsdb'
*Verdi* : Begin traversing the scope (UQ_tb), layer (0).
*Verdi* : End of traversing.
First challenge
CHIP 0 : at time              1155000, chall_in = 01, response = 4b, ready = 1
CHIP 1 : at time              1155000, chall_in = 01, response = 66, ready = 1
CHIP 2 : at time              1155000, chall_in = 01, response = 6c, ready = 1
CHIP 3 : at time              1155000, chall_in = 01, response = 53, ready = 1
CHIP 4 : at time              1155000, chall_in = 01, response = 01, ready = 1
CHIP 5 : at time              1155000, chall_in = 01, response = 2e, ready = 1
CHIP 6 : at time              1155000, chall_in = 01, response = 4c, ready = 1
CHIP 7 : at time              1155000, chall_in = 01, response = 74, ready = 1
CHIP 8 : at time              1155000, chall_in = 01, response = 13, ready = 1
CHIP 9 : at time              1155000, chall_in = 01, response = 57, ready = 1
UQ = 0.46
Second challenge
CHIP 0 : at time              3205000, chall_in = c1, response = 4e, ready = 1
CHIP 1 : at time              3205000, chall_in = c1, response = 77, ready = 1
CHIP 2 : at time              3205000, chall_in = c1, response = 13, ready = 1
CHIP 3 : at time              3205000, chall_in = c1, response = 3c, ready = 1
CHIP 4 : at time              3205000, chall_in = c1, response = 06, ready = 1
CHIP 5 : at time              3205000, chall_in = c1, response = 62, ready = 1
CHIP 6 : at time              3205000, chall_in = c1, response = 25, ready = 1
CHIP 7 : at time              3205000, chall_in = c1, response = 2d, ready = 1
CHIP 8 : at time              3205000, chall_in = c1, response = 12, ready = 1
CHIP 9 : at time              3205000, chall_in = c1, response = 76, ready = 1
UQ = 0.44
Third challenge
CHIP 0 : at time              5165000, chall_in = 2c, response = 7f, ready = 1
CHIP 1 : at time              5165000, chall_in = 2c, response = 38, ready = 1
CHIP 2 : at time              5165000, chall_in = 2c, response = 40, ready = 1
CHIP 3 : at time              5165000, chall_in = 2c, response = 1a, ready = 1
CHIP 4 : at time              5165000, chall_in = 2c, response = 02, ready = 1
CHIP 5 : at time              5165000, chall_in = 2c, response = 4c, ready = 1
CHIP 6 : at time              5165000, chall_in = 2c, response = 69, ready = 1
CHIP 7 : at time              5165000, chall_in = 2c, response = 48, ready = 1
CHIP 8 : at time              5165000, chall_in = 2c, response = 76, ready = 1
CHIP 9 : at time              5165000, chall_in = 2c, response = 4f, ready = 1
UQ = 0.44
Fourth challenge
CHIP 0 : at time              7215000, chall_in = 01, response = 4b, ready = 1
CHIP 1 : at time              7215000, chall_in = 01, response = 66, ready = 1
CHIP 2 : at time              7215000, chall_in = 01, response = 6c, ready = 1
CHIP 3 : at time              7215000, chall_in = 01, response = 53, ready = 1
CHIP 4 : at time              7215000, chall_in = 01, response = 01, ready = 1
CHIP 5 : at time              7215000, chall_in = 01, response = 2e, ready = 1
CHIP 6 : at time              7215000, chall_in = 01, response = 4c, ready = 1
CHIP 7 : at time              7215000, chall_in = 01, response = 74, ready = 1
CHIP 8 : at time              7215000, chall_in = 01, response = 13, ready = 1
CHIP 9 : at time              7215000, chall_in = 01, response = 57, ready = 1
UQ = 0.46
```

- Challenges = **01,c1,2c,01**, using **10 different RO PUF** and **3 groups of challenges** and 1 for checking stability.
- Average UQ = (46+44+44)/3 = **44.67%**



Analysis

- **FRO PUF**'s UF and UQ is better (more close to 50%), but have **small range** of **output domain** (`0x00==0xff`, `0x01==0xfe`,...(**symmetry**)), and also **different challenge may give same response** in same chip
- **RO PUF** can still work even the input challenge is **0x00 or 0xff**, and not having same output in above experiments.
- **RO PUF** have **more range** of challenges domain and can still provide **different response**, while **FRO PUF** have better score in both **UF and UQ**.
    - **BUT** it may because of using too less sample to calculate the result
    - If sample large enough, both may reach 50% of UF and UQ
- **RO PUF** have **simple structure**, **easy** to **design** and **low cost** (only invertor), while **FRO PUF** have a **complex design** than **RO PUF**, and use multiple type of gate to implement(NOT,AND) so is more **costly**
- **RO PUF** are **sensitive** to environmental variations and noise sources such as **temperature and voltage**, '**confidence**' of a **0/1** measurement are low