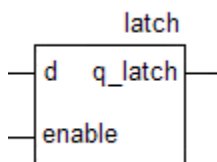


## Modelarea și simularea latch-ului și bistabilului tip D

Diferența dintre cele două circuite se datorează modului lor diferit de funcționare. Astfel, latch-ul este caracterizat de comutarea sa doar pe palierul semnalului de autorizare (pe *High* sau pe *Low*), în vreme ce bistabilul tip D comută doar pe frontul pozitiv al semnalului de clock.

Denumirea de latch provine din faptul că prin comanda **enable** circuitul poate fi "zăvorât" fie în starea "sus", caz în care ieșirea  $q_{latch} = 1$ , fie în starea "jos", caz în care  $q_{latch} = 0$ .

a) **Schema bloc** a unui **latch tip D** activ pe palierul de "1" al semnalului **enable** (de autorizare) este prezentată mai jos, unde *d* este intrarea de date iar  $q_{latch}$  ieșirea de date.



b) **Tabelul de adevăr** reflectă comportamentul latch-ului: când **enable** ia valoarea 1, valoarea semnalului aplicat la intrarea *d* este copiată la ieșirea  $q_{latch}$  a latch-ului, altfel valoarea ieșirii rămâne neschimbată.

<i>enable</i>	<i>Q</i>
0	Q
1	D

c) **Codul verilog implementat pentru modelarea latch-ului de tip D**

Latch-ul tip D destinat simulării a fost denumit generic **DUT** (*Device Under Test*). Descrierea modulului său în cod Verilog este prezentată mai jos. Modulul reprezintă o descriere comportamentală. Numele **latch** de identificator al modulului este și numele fișierului **latch.v** (cu extensia „v”) în care acesta va fi salvat pe disc (procedeu recomandat dar nu și obligatoriu).

```

1  module latch (d, enable, q_latch);
2
3  input d, enable;
4  output q_latch;
5
6  reg q_latch;
7
8  always @ (d or enable)
9      if (enable)
10         //begin
11             q_latch <= d;
12         //end
13  endmodule
  
```

Din procedura de simulare a fost extrasă diagrama prezentată în pagina următoare. Diagrama pune în evidență funcționarea latch-ului modelat cu modulul **latch** descris mai sus.

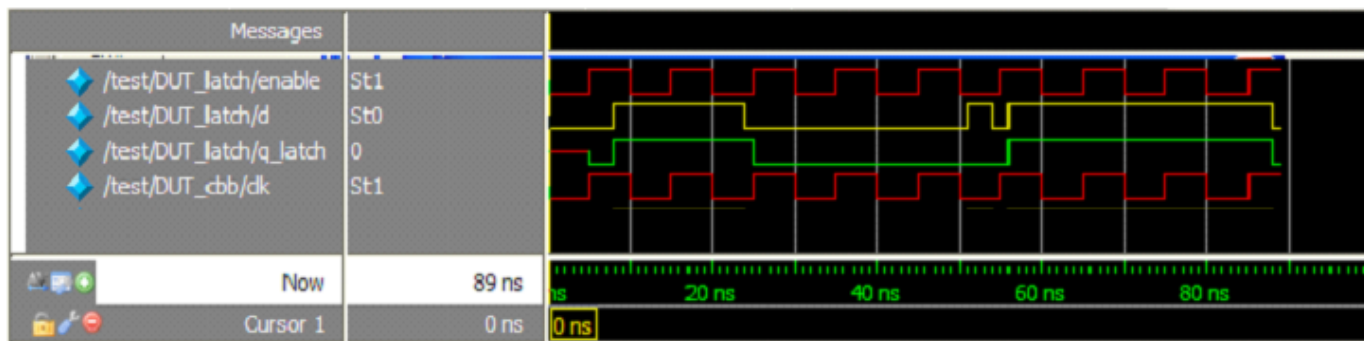
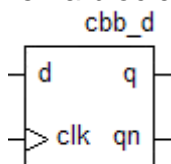


Diagrama simulării latch-ului tip D

- d) Schema bloc a **bistabilului tip D**, activ pe frontul pozitiv al semnalului **clk** (de ceas):



- e) **Tabelul de adevăr** descrie comportamentul bistabilului de tip D: pe frontul pozitiv al semnalului **clk** de ceas, valoarea prezentă pe intrarea **d** este transferată la ieșirea **q**, adică ieșirea bistabilului stochează (memorează) valoarea prezentă la intrare. Pe palierul semnalului de clock nu are loc acest transfer, q post-tact fiind același cu q ante-tact.

<b>clk</b>	<b>q<sup>+</sup></b> (post-tact)
pe palier (0 sau 1)	q (ante-tact)
front +	d

- f) **Codul verilog scris pentru modelarea bistabilului de tip D**

Bistabilul tip D (Device Under Test -DUT) este descris în modulul **cbb\_d**. Acest modul reprezintă descrierea comportamentală a bistabilului și se va salva în fișierul **cbb\_d.v**. Bistabilul prezintă intrările **d** și **clk** și ieșirile **q** și **qn** (pentru q negat).

```

1  module cbb_d (d, clk, q, qn);
2  input d, clk;
3  output q, qn;
4
5  reg q, qn;
6
7  always @(posedge clk)
8  begin
9      q <= d;
10     qn <= ~d;
11 end
12 endmodule

```

Din procedura de simulare a fost extrasă diagrama de mai jos. Această diagramă evidențiază funcționarea circuitului basculant bistabil tip D descris în modulul **cbb\_d**.

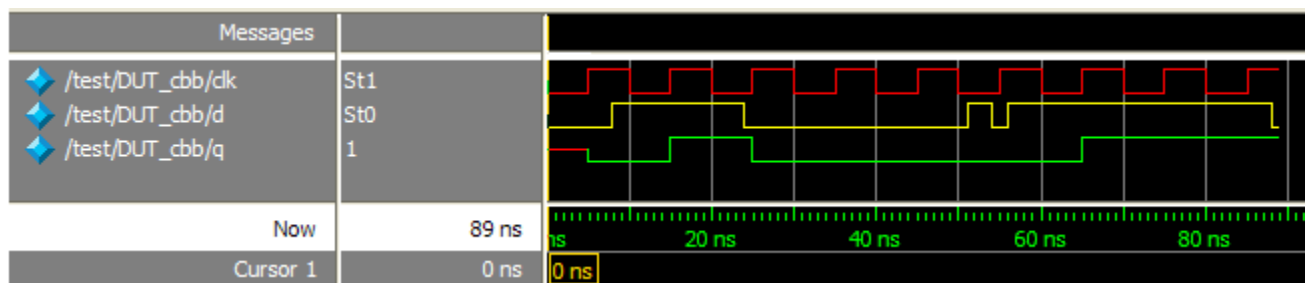


Diagrama simulării unui CBB tip D

### g) Codul Verilog scris pentru modelarea generatorului de stimuli TestBench (TB)

Pentru generarea stimulilor necesari testării, anume **d** (date) și **clk** (ceas) sau **enable** (autorizare latch) s-a creat modulul denumit **test\_bench** prezentat mai jos. Blocul acestui generator de stimuli a fost denumit generic TestBench (TB). Conținutul modulului se va salva într-un fișier denumit **test\_bench.v** (sub același nume cu cel al modulului).

```

1  module test_bench (d, clk);
2  output d, clk;
3
4  reg d, clk;
5
6  initial
7  begin
8      clk = 0;    //initializare semnal clk de clock
9      d = 0;      //initializare semnal d intrare date
10     #8  d = 1;   //intarziere cu 8 unitati timp a asignerii d=1
11     #14 d = 0;   //intarziere cu 8+14= 22 unitati timp
12     #27 d = 1;   //intarziere cu 22+27= 49 unitati timp
13     #3  d = 0;   //intarziere cu 49+3= 52 unitati timp
14     #2  d = 1;   //intarziere cu 52+2= 54 unitati timp
15     #32 d = 0;   //intarziere cu 54+32= 86 unitati timp
16     #1 $stop;
17 end
18
19 always #5 clk = ~clk; //comutare clock tot la 5 unitati de timp
20 endmodule

```

### h) Simularea concomitentă a latch-ului tip D și a bistabilului tip D

Diagrama de simulare ce va fi ridicată, în laborator, folosind programul ModelSim permite realizarea unei comparații între modul de funcționare al latch-ului tip D și cel al bistabilului de tip D. Din acest motiv s-a folosit instanțierea ambelor module comportamentale (**latch** și **cbb\_d**) în același modul (**test**) de testare.

În modulul **test** (salvat sub numele **test.v**) a fost instanțiat și modulul generatorului de stimuli **test\_bench**. Modulul **test** este o descriere structurală a conexiunilor dintre cele trei module (**latch**, **cbb\_d** și **test\_bench**) instanțiate. Instanțele celor trei module au fost

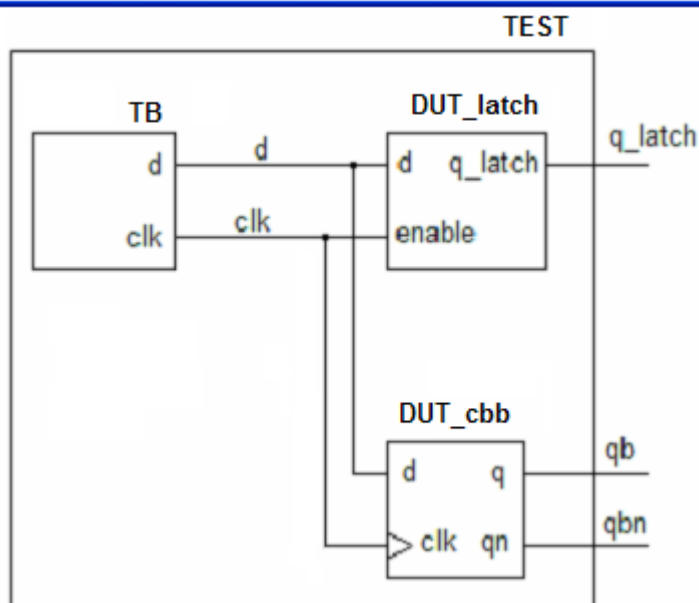
#### CLP Laborator 4- Modelarea și simularea latch-ului și bistabilului tip D

botezate DUT\_latch, DUT\_cbb și respectiv TB. După fiecare instanță sunt înscrise, între paranteze, grupurile **.port(conductor)**, unde **port** (precedat de un punct) este numele unui port al modulului instanțiat iar **conductor** este numele conductorului la care se leagă portul respectiv. Conductoarele de legătură sunt declarate în declarația **wire** a modulului **test**.

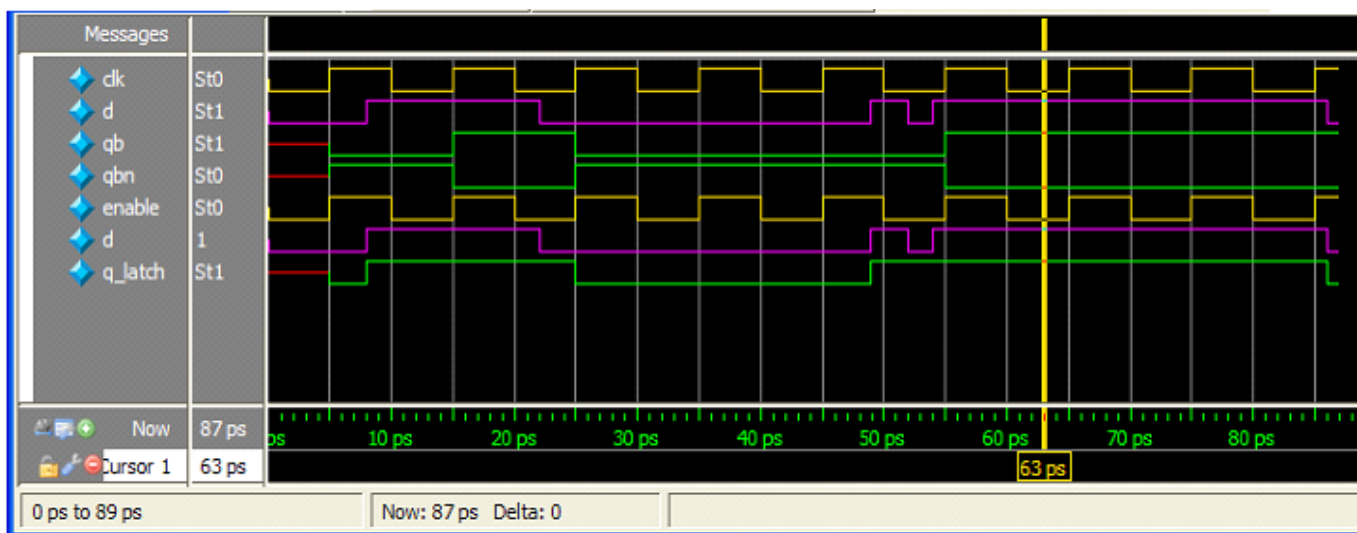
```

1  module test ();
2  wire d, clk, qb, qbn, q_latch;
3
4  test_bench TB (.d(d), .clk(clk) );
5  latch DUT_latch (.d(d), .enable(clk), .q_latch(q_latch) );
6  cbb_d DUT_cbb (.d(d), .clk(clk), .q(qb), .qn(qbn) );
7
8  endmodule

```



Schema bloc a conexiunilor dintre componentele modelate



Digrama semnalelor extrasă din panoul **wave** al programului ModelSim folosit la simulare

ns	enable	d	q_latch	clk	d	q
0	St0	St0	x	St0	St0	x
5	St1	St0	0	St1	St0	0
8	St1	St1	1	St1	St1	0
10	St0	St1	1	St0	St1	0
15	St1	St1	1	St1	St1	1
20	St0	St1	1	St0	St1	1
24	St0	St0	1	St0	St0	1
25	St1	St0	0	St1	St0	0
30	St0	St0	0	St0	St0	0
35	St1	St0	0	St1	St0	0
40	St0	St0	0	St0	St0	0
45	St1	St0	0	St1	St0	0
50	St0	St0	0	St0	St0	0
51	St0	St1	0	St0	St1	0
54	St0	St0	0	St0	St0	0
55	St1	St0	0	St1	St0	0
56	St1	St1	1	St1	St1	0
60	St0	St1	1	St0	St1	0
65	St1	St1	1	St1	St1	1
70	St0	St1	1	St0	St1	1
75	St1	St1	1	St1	St1	1
80	St0	St1	1	St0	St1	1
85	St1	St1	1	St1	St1	1
88	St1	St0	0	St1	St0	1

Lista semnalelor extrasă din panoul **list** al programului de simulare ModelSim

În lista **list** de mai sus se observă evoluția valorilor semnalelor de intrare și de ieșire pe parcursul duratei de 88 ns a procesului de simulare. Unitatea de timp utilizată în acest caz de simulator este ns (nano-secunda).

Se poate observa că semnalul **clk** schimbă de stare din 5 în 5 unități de timp (conform comenzii de comutare **#5 clk <= ~clk**). La fiecare tranziție a **clk** din 0 în 1 (de front crescător), starea intrării **d** a bistabilului este copiată la ieșirea **q** a acestuia.

În cazul latch-ului, câtă vreme **enable** este pe palierul High (1 logic), ieșirea **q\_latch** copiază valoarea prezentă la intrarea **d**.

Întocmit,  
Coordonator lucr. laborator  
Ing. Hurubeanu Ștefan Valeriu

Brașov, 20 Martie 2011