Крайни автомати - примери

[Edge Detector FSM](#_fuyzlifxrjrr)

[edge\_detector.sv](#_y2s56tuthe7j)

[Edge Detector + Debouncer](#_sz4riw1hjcux)

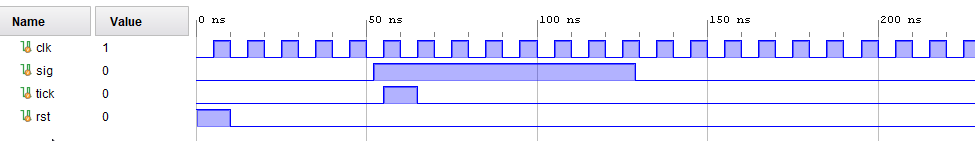
[Contact bounce](#_l3fu9qvsik9m)

[edge\_detector\_delay.sv](#_cq2xbf85ndi9)

# Edge Detector FSM

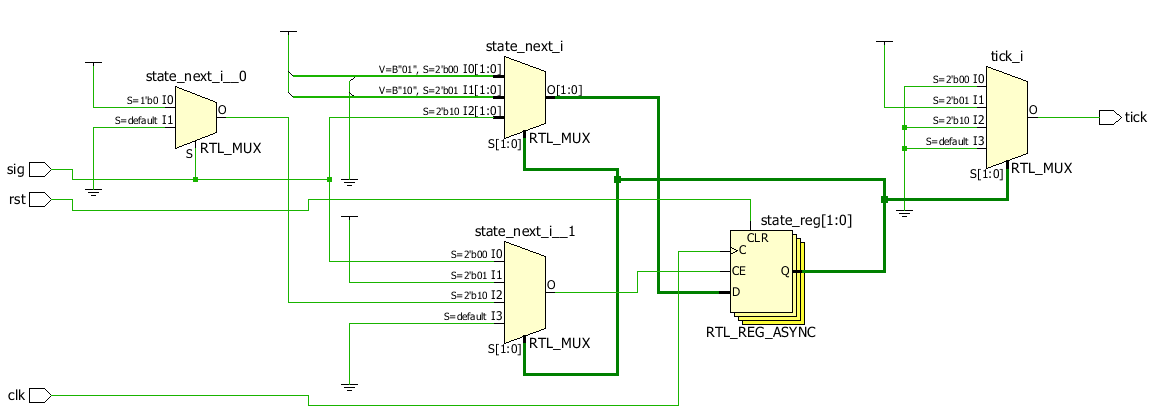
Схема за изработване на единичен импулс (с продължителност равна на периода на тактов сигнал).

Използва се за синхронизиране на входни сигнали с такта на последователни схеми.

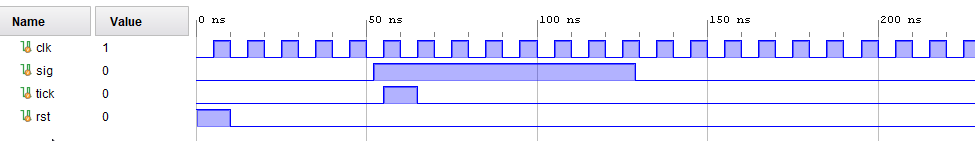


## edge\_detector.sv

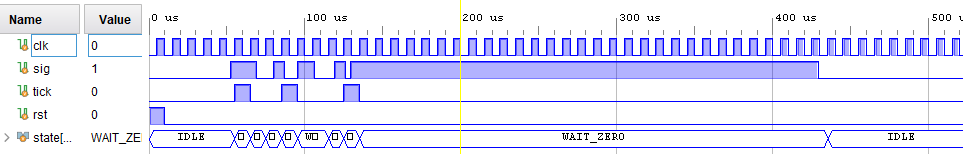
| module edge\_detector(  input clk, rst, sig,  output logic tick );  enum logic [1:0] {IDLE, EDGE, WAIT\_ZERO} state, state\_next;  always\_ff @(posedge clk, posedge rst)  if (rst) state <= IDLE;  else state <= state\_next;  always\_comb begin  state\_next = state;  tick = 0;  case (state)  IDLE:  if (sig) state\_next = EDGE;  EDGE: begin  tick = 1;  state\_next = WAIT\_ZERO;  end  WAIT\_ZERO:  if (~sig) state\_next = IDLE;  endcase  end  endmodule | edge_detector.png |
| --- | --- |

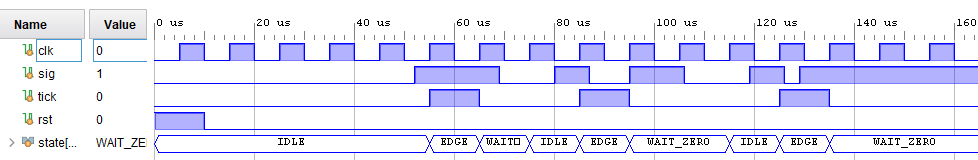


**edge\_detector\_test1**



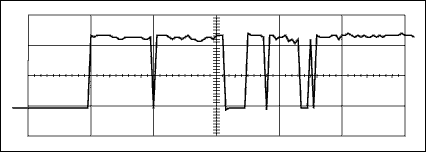
**edge\_detector\_test2**



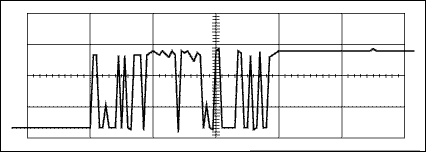


# Edge Detector + Debouncer

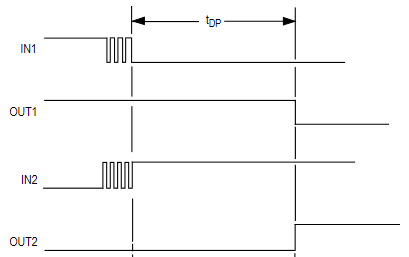
## Contact bounce



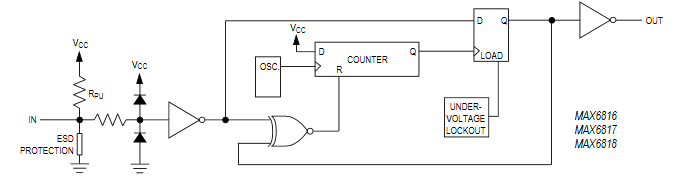
*This rising-edge switch bounce for a small pushbutton switch shows an approximate 5ms bounce interval that includes 10 transitions.*

**

*Another rising-edge switch bounce (for a 5A contact relay) shows an approximate 5.5ms bounce interval that includes 20 full-amplitude transitions and a few smaller ones.*

**

tDP = 20-40ms

**

*Source:*

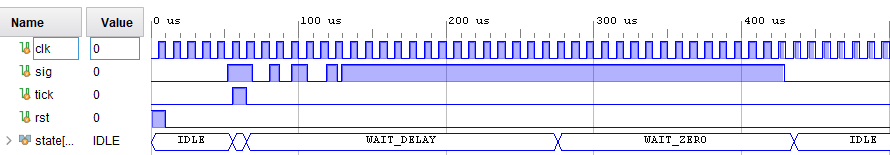
[*http://pdfserv.maximintegrated.com/en/an/AN764.pdf*](http://pdfserv.maximintegrated.com/en/an/AN764.pdf)

[*https://datasheets.maximintegrated.com/en/ds/MAX6816-MAX6818.pdf*](https://datasheets.maximintegrated.com/en/ds/MAX6816-MAX6818.pdf)

## edge\_detector\_delay.sv

| module edge\_detector\_delay #(parameter DELAY=10)(  input clk, rst, sig,  output logic tick );  enum logic [1:0] {IDLE, EDGE, WAIT\_ZERO} state, state\_next;  **logic [5:0] timer, timer\_next;**  always\_ff @(posedge clk, posedge rst)  if (rst) begin  state <= IDLE;  **timer <= '0;**  end  else begin  state <= state\_next;  **timer <= timer\_next;**  end  always\_comb begin  state\_next = state;  timer\_next = timer;  tick = 0;  case (state)  IDLE:  if (sig) state\_next = EDGE;  EDGE: begin  tick = 1;  state\_next = WAIT\_DELAY;  timer\_next = DELAY;  end  WAIT\_DELAY: begin  if (timer) timer\_next = timer - 1;  else state\_next = WAIT\_ZERO;  end  WAIT\_ZERO:  if (~sig) state\_next = IDLE;  endcase  end  endmodule  timer -> [register] -> timer\_next | edge_detector_delay.png |
| --- | --- |

*edge\_detector\_delay / edge\_detector\_delay\_test*

**