Simulation : 0

Event Driven simulation

- Uses standard event driven logic simulation algorithms.
  - Described best with simulation of structural models.
  - Behavioural models use process as an element.

Minimize the work done by the simulator by:

  - evaluating the circuit state only when an input changes.
  - can predict that an output might change if inputs change.
  - logic function dictates whether a change will occur.
  - delay information tells us when it will change.
  - re-evaluate elements only when the possible changes occur

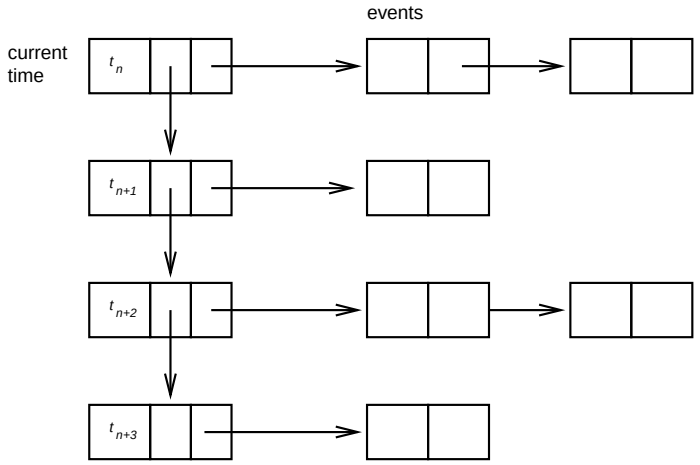
By following these possible events through the circuit we minimize the work done by the simulation. Only elements that change need to be evaluated.

Simulation : 1

Delays through elements are defined in terms of integer times, eg. picoseconds or nanoseconds.

- Simulation time is incremented in discrete steps so sometimes
  - more than one element may need evaluating,
  - no elements may need evaluating.
  - changes in element outputs will cause other inputs to change ...

Therefore we need to maintain a list of signals and when they change (events).



Simulation : 2

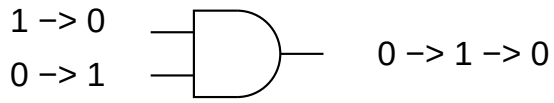
Event Scheduling:

- An event will only be scheduled if the new value is different to the current value.

But:

- What if two events are scheduled for the same signal?
  - and have the same value?
  - or have different values?

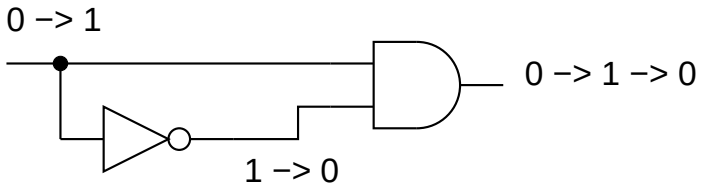
This could lead to a zero width spike one delay later. Or worse if the rise and fall times for the changes are different as this could lead to an output pulse whose presence, or absence, depended on the ordering of the events.



Event Scheduling:

Zero delay gates can also cause problems in simulation. A circuit with feedback may result in an infinite loop as transactions that cause events on the outputs will be scheduled for evaluation in the current cycle.

The order of evaluation can have an effect.



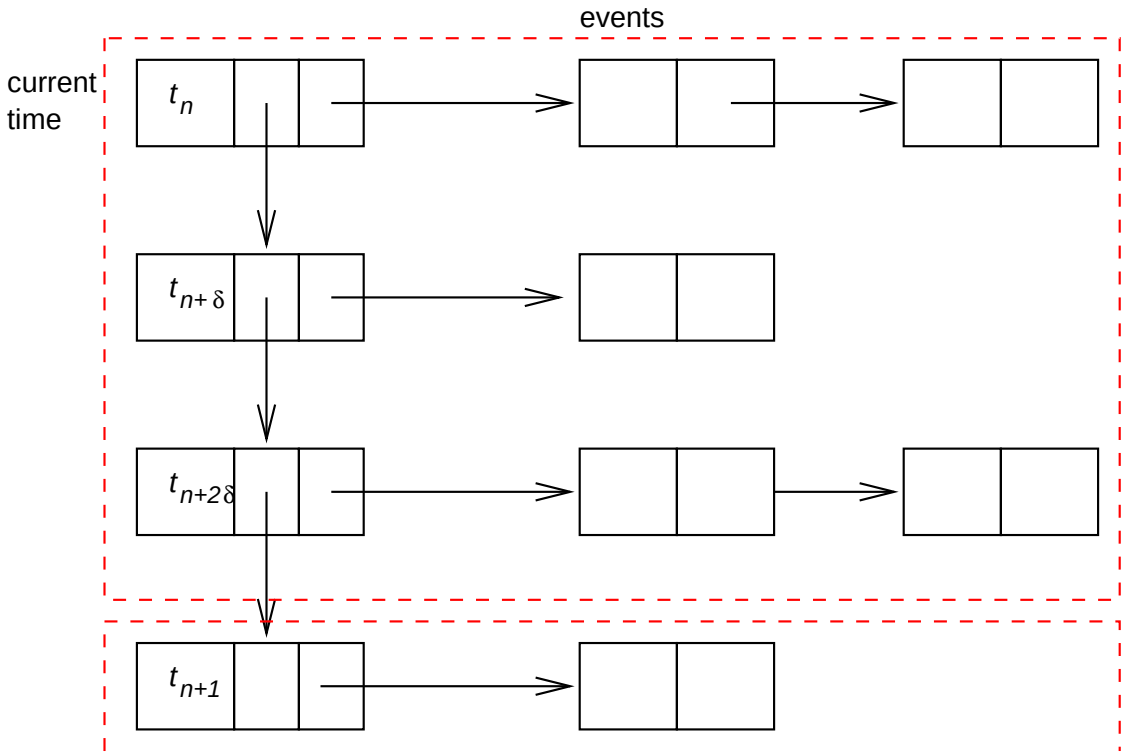
- Transaction of 0 -> 1 on AND causes evaluation and event of 0 -> 1 on output.
- Transaction of 0 -> 1 on NOT causes evaluation and event of 1 -> 0 on output.

this triggers a transaction on the other AND input so

- Transaction of 1 -> 0 on AND causes evaluation and event of 1- > 0 on output.

Evaluating the NOT first results in a different event list for the output of the AND gate.

Delta time:

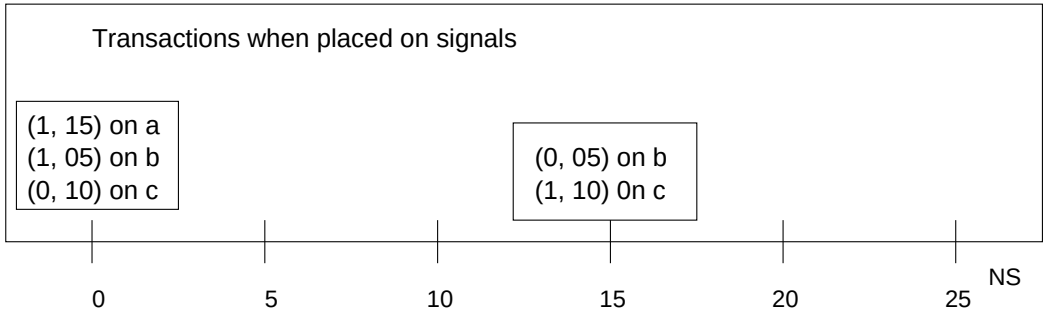


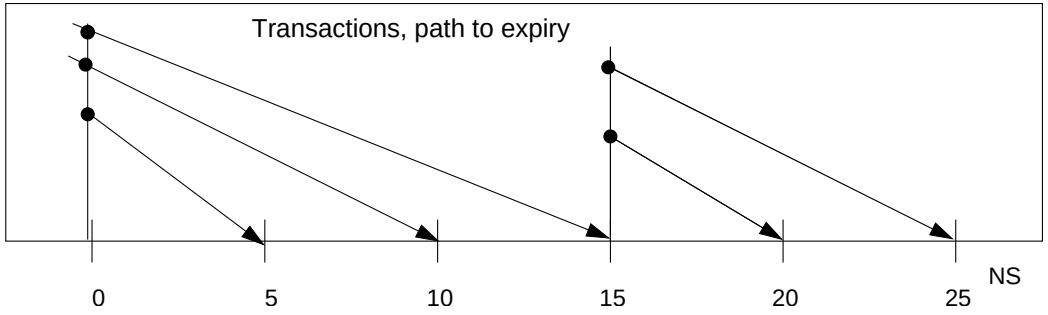
Delta time permits consistent evaluation of zero delays as they are modelled as delta delays. Any number of delta delays can occur in one time step.

Simulation : 5

architecture demo of example is

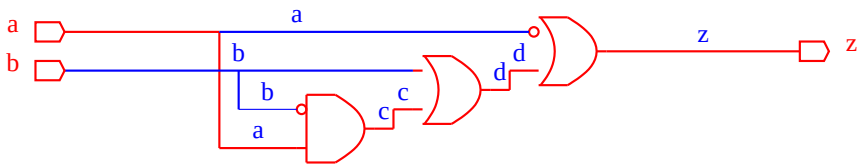
```
signal a, b, c : bit
begin
  a <= '1' after 15 ns ;
  b <= NOT a after 5 ns ;
  c <= a after 10 ns ;
end demo;
```



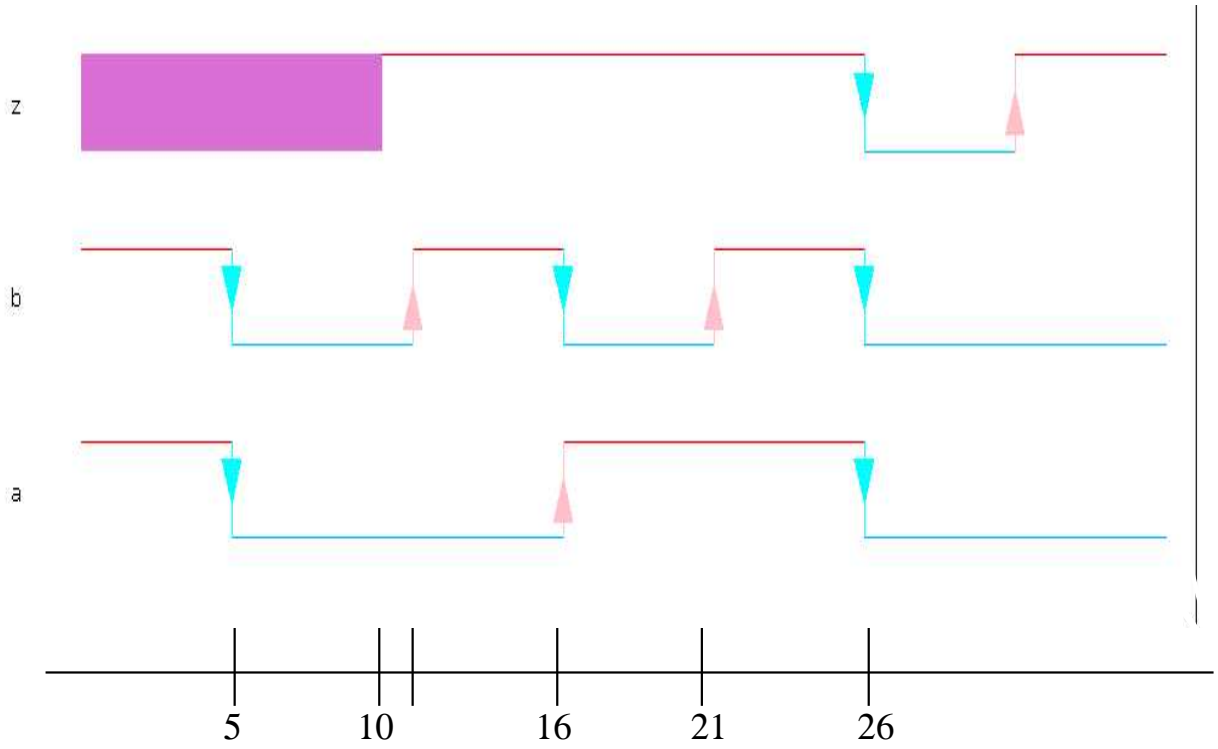


Simulation : 5+d

```
entity q6 is

  port (
    vdd: in bit;
    vss: in bit;
    A : in  bit;
    B : in  bit;
    Z : out bit);

end q6;

architecture dataflow of q6 is
  signal C,D : bit;
begin  -- dataflow
  Z <= not A or D  after 5ns;
  C <= A and not B after 5ns;
  D <= C or B      after 5ns;
end dataflow;
```

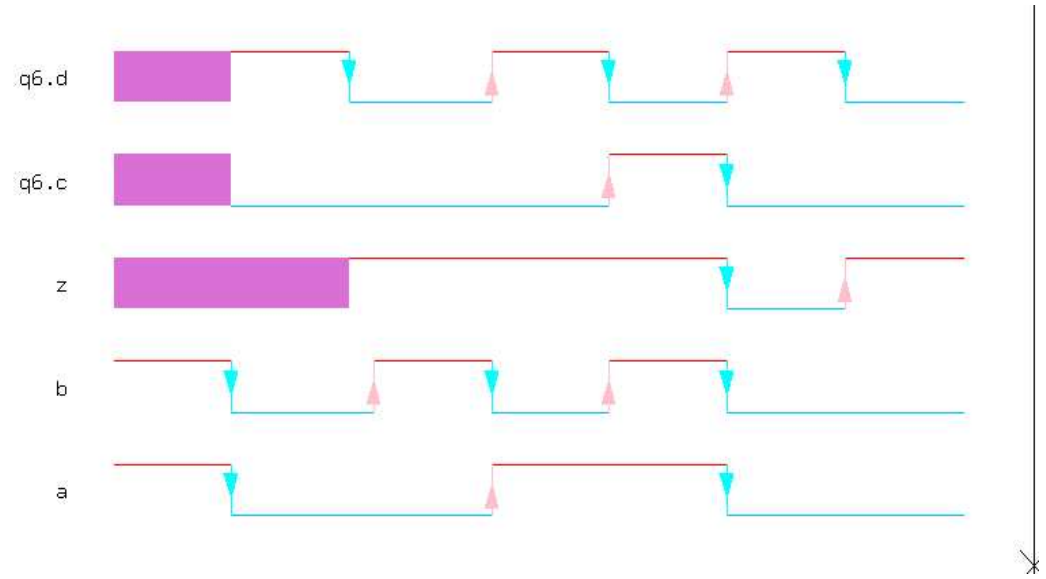


Simulation : 6



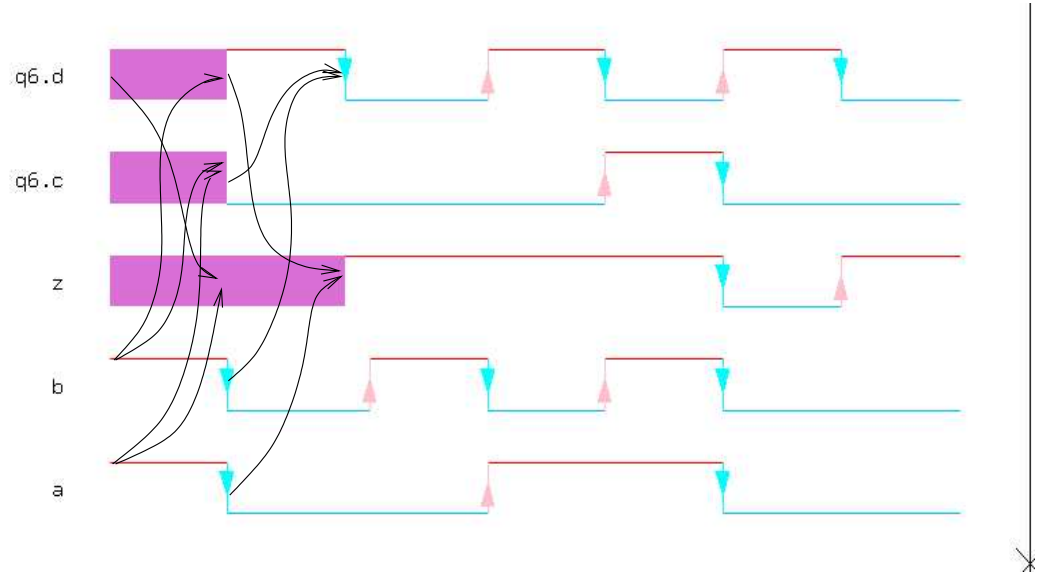
Z <= not A or D after 5ns;
C <= A and not B after 5ns;
D <= C or B after 5ns;

Simulation : 7



Z <= not A or D after 5ns;
C <= A and not B after 5ns;
D <= C or B after 5ns;

Simulation : 7+d



Z <= not A or D after 5ns;
C <= A and not B after 5ns;
D <= C or B after 5ns;