

Gate level description of D-type flip-flop and timing issues.

This example provides an opportunity to explore some of the hardware specific issues that arise when dealing with gate level descriptions of sequential logic, ie. circuits that have feedback in them. This can be a particular problem in simulation using the Alliance simulation tool, Asimut, especially if the timing information is missing or incorrect. Timing information can be present in the behavioural description through the use of the `after` clause. The simulator can be provided with additional timing and delay information through the use of the flags `-fd n` or `-dd n`. This will allow the simulation of structural models where feedback is causing problems, see `asimut man` pages for the details. Finally, timing information is also present in the pattern files and has a significant effect on the behaviour and performance of the design being simulated.

The circuit provided is that of a D-type flip-flop with `preset` and `clear` inputs and is shown in *figure 1*. This circuit is taken from the commercial design for a positive-edge-triggered D flip-flop such as the 74LS74. With a design such as this it is crucial that the inertial delays are represented correctly and attention is paid to the input setup times if the design is to be simulated. The accompanying pattern files and results demonstrate this.

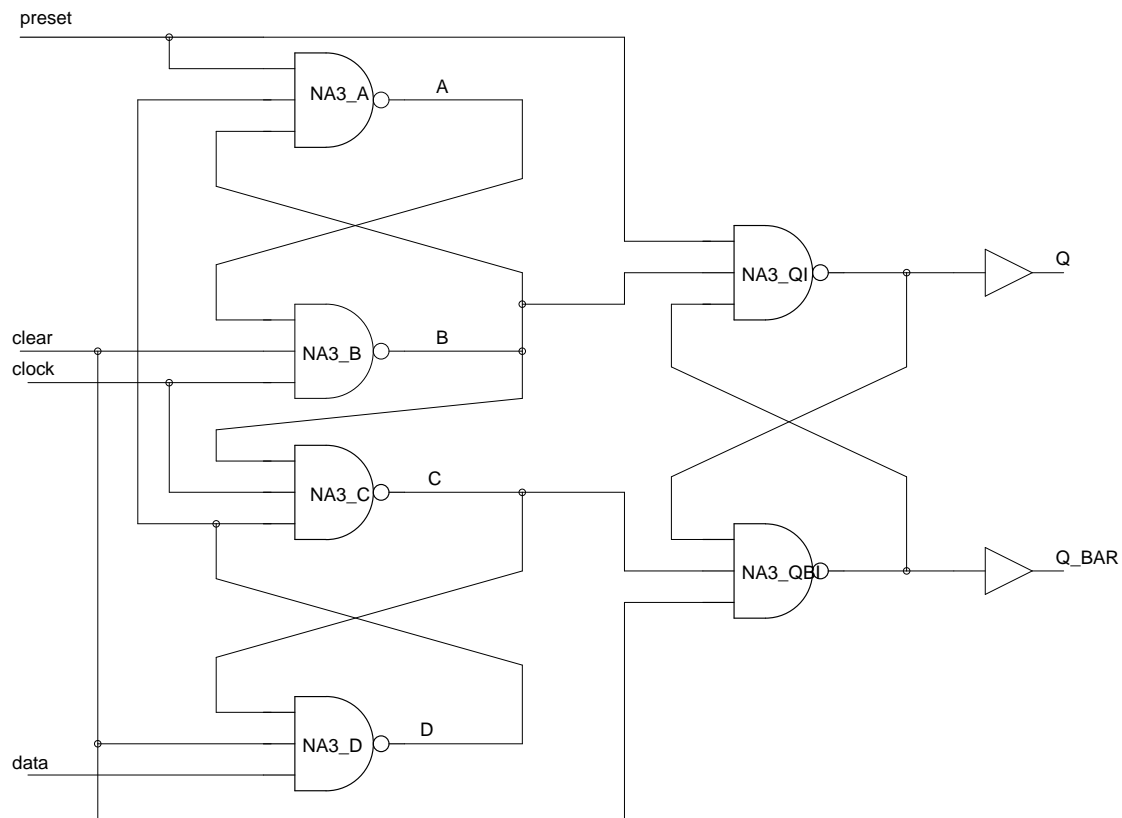


figure 1

- Behavioural Descriptions

Base level behavioural description is in `d-type.no-spy.vbe`. This uses the `after` clause to control timing. It can be simulated with the pattern file `d-type.b.2ns.in.pat`. The results, in `d-type.b.2ns.out.pat`, show a number of glitches and other errors. Note that the inputs are in transition on the rising clock edge and that both `preset` and `clear` are shown changing at the same time. The pattern file uses `4x2nS` lines per clock cycle

A behavioural description using the `spy` functionality, in `d-type.vbe`, when simulated with the pattern file `d-type.in.6ns.pat` shows the changing values on the internal signals. This

is quite revealing in that it shows the complexity of the internal signals and instability in the output. This file uses the same input pattern as the previous one, but with the addition of the internal signals. Note that the resolution of the output file is at 1nS intervals.

- Q? What conclusions can be drawn from these results?
- Q? What happens if `preset` and `clear` are active at the same time?
- Q? How long should inputs be stable for before the d-type is clocked?
- Q? Develop a test pattern for the behavioural model that ensures that inputs are stable before the rising clock-edge. What are the results? Try this with both the `spy` and `no-spy` models.
- Structural Descriptions
- 6ns `d-type.s.6ns.in.pat` is a coarse grained pattern file with $4 \times 6nS$ lines per clock cycle. Note that not all of the input signals change on the clock edges in this pattern file. This pattern file is the same as the 6ns behavioural pattern file but without the `spy` signals. It also demonstrates problems with the output.
- 1ns This pattern file, `d-type.s.1ns.in.pat`, uses a fine grained approach. There are $24 \times 1nS$ pattern lines per clock cycle. This allows experimentation with the setup and hold times for this flip-flop. All of the input signals in this example are stable for several nanoseconds before the rising clock-edge. The outputs show no sign of instability.

Comments and Problems

Although the behavioural vhd description will simulate perfectly well with all the pattern files given, it cannot be converted to a structural description using either `scmap` or `bop` and then `scmap`. This includes using a constraints file `filename.lax`, (see `man lax` for details), to control the level of optimisation, input delays, retaining internal signals etc. Both routes to structural description result in segmentation faults and further investigation is under way.

A manually derived structural description, such as `d-type.vst`, will simulate perfectly well provided that either

- input signals do not change on clock edges
- or
- if they do, then you must use the `-fixeddelay value` or `-fd value` flag with the simulator as in

```
asimut -fd 1ns dodgy_struct badtestin testout
```

- Q? What are the setup and hold times, the propagation delay and the maximum clock rate for the 74LS74 ic?
- Q? Develop a test pattern that models the setup, hold and clock rates for the 74LS74, apply it to the given structural and behavioural models.
- Q? Do a timing analysis on the structural model using `tas`. You will need to convert the structural description to alliance netlist format. This contains the physical characteristics of the implementation technology, eg. standard cells. The timing information is derived from the capacitance of the inputs and outputs of the cells.

```
x2y vst al mystructuralfile newfile
```

This will create a file with a `.al` extension. There are only five licenses for the timing analysis tool, however the license is only required while the software is actually running, a matter of seconds usually. There are two tools for viewing the generated timing reports, `etas` and `xtas`.