

This is a really short tutorial that demonstrates the entire process of compiling and running a simulation with NC-Verilog & Specman (using compiled specman).

Consider the following sample design - **my\_xor.v**:

{code}

```
module my_xor (A,B,Y);
```

```
    input A;
```

```
    input B;
```

```
    output Y;
```

```
    assign Y = A ^ B;
```

```
endmodule
```

{/code}

A very complex DUT as you may have noticed.

Now let's add the environment - **env.e**:

```
{code}
```

```
extend sys {
```

```
run() is also {
```

```
    start my_test();
```

```
};
```

```
my_test()@sys.any is {
```

```
    out( ===== START ,sys.time);
```

```
    wait delay (1000 ns);
```

```
    '~/my_xor/A' = 1;
```

```
'~/my_xor/B' = 0;
```

```
wait delay (1000 ns);
```

```
'~/my_xor/A' = 0;
```

```
wait delay (1000 ns);
```

```
out( ===== END ,sys.time);
```

```
};
```

```
};
```

```
{/code}
```

---

It's a just a quick demo env so excuse the coding style. And now for the fun part - Let's simulate our little environment with the DUT -

Open a fresh terminal and make sure you got the 2 files at your current dir (copy & paste into my\_xor.v and env.e) and follow these steps:

1. First , compile the e code into a shared library so that an executable file is created and also a

corresponding esv file:

```
%sn_compile.sh -shlib -exe env.e
```

2. Use the newly created executable file (env) to create verilog stubs file:

```
%./env -c write stubs -ncvlog
```

3. Compile the verilog code plus the newly created stub file (specman.v):

```
%ncvlog -MESSAGES -UPDATE my_xor.v specman.v
```

4. Elaborate the design:

```
%ncelab_specman specman my_xor -snapshot my_xor -access +rwc
```

5. Now you have to tell Specman where to find the compilation results

```
%setenv SPECMAN_DLIB ./libsn_env.so
```

6. That's is - now run the simulation:

```
%specrun -p test ncsim -nbasync my_xor
```

As usual, read the manual to get the full picture but the example above should give you the hang of it and may be used as reference. And if you were wondering if there was a single script solution for this, well - there is. Cadence recently released the irun script which should take care of everything you always wanted from your simulator but were afraid to ask. Quite a catchy name for a script, don't you think?