Traffic Light Testbench Guidance

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachgebiet Integrated Electronic Systems
Prof. Dr.-Ing. Klaus Hofmann
Date: Winter semester 01.02.2020

Contents

1 Introduction

This document explains the logs that you could encounter during your design debugging. To get the most out of this testbench you should clearly annotate the meaning of the states in your design so that you can follow the testbench hints to correct your code.

2 Debugging and Logs

2.1 Reset state requirement

One repuirement to use the testbench is the reset state in your design should be the setup state of **Hauptstraße Nebenstraße Rot** which is followed by **Hauptstraße Rot Gelb**. Aswell the timer should be set to 1 in the reset state. Otherwise, the following error will occur:

```
# Der Timer sollte im Zustand Hauptstrasse Nebenstrasse Rot Setup (reset Zustand) auf 1 eingestellt sein
```

Listing 1: Reset state error

2.2 Errors of green light states

In the states **Hauptstraße Grün** and **Nebenstraße Grün**, the testbench will turn on the **hs_f_an** and **ns_f_an** signals at different times to test the response of your design. In case these signals are turned on and the output **Q** of the timer is larger than 5, you should set the timer to 5. If something is wrong at this step, the following error will occur:

```
# Testfall 1 ausfuehren:
# Testen die Ampel mit hs_f_an = 1 und ns_f_an = 1 in der 15. Sekunde der gruenen Zustaende.
# Testen das ns_f_an signal...
# Timer fuer Hauptstrasse Gruen Signal Kommt ist nicht eingestellt, wenn Q groesser als 5 ist.
```

Listing 2: Timer Setup error when Q larger then 5

when the output **Q** of the timer is smaller than 5, the timer parameter should not be changed. If something is wrong at this step, the following error will occur:

```
# Testen die Ampel mit hs_f_an = 1 und ns_f_an = 1 in der 4. Sekunde der gruenen Zustaende.
# Testen das ns_f_an signal...
# Fehler im Zustand Hauptstrasse Gruen Signal Kommt.
# Der Timer-Parameter sollte nicht geaendert werden, wenn Q kleiner als 5 ist.
```

Listing 3: Timer Setup error when Q smaller then 5

To solve the above errors, you should check signals **init** and **load** in relevant states.

2.3 Stuck state error

The testbench will check all the states lasting more than 1 second. In case, the output of your design doesn't change for a long time, this error will pop up as in the following example:

```
# Testfall 0 ausfuehren:
# Testen die Ampel mit hs_f_an = 0 und ns_f_an = 0
# Fehler im Zustand:          Hauptstrasse Rot Gelb Setup
# ------------------------------Hinweise--------------------------------
# Vorheriger Zustand, der den Test besteht:       Hauptstrasse Nebenstrasse Rot
# Ueberpruefen Sie ab besteheden Zustand, ob Sie ihr Programm korrekt auf dem naechsten Zustand eingestellt wurde.
```

Listing 4: Stuck state error

The reason for that may be that you have some state transition flaws, leading to your design unable to move to the next state. The testbench provides you the closest state your design responds exactly. From this state, you can check your transitions and find out errors.

2.4 Signal mismatch error

In most cases you will get the error as in the following example:

```
# Testfall 0 ausfuehren:
# Testen die Ampel mit hs_f_an = 0 und ns_f_an = 0
# Testen das ns_f_an signal...
# Fehler im Zustand:                    Hauptstrasse Gruen
# Signal  hs_f_rt ist  0 !=  1
# ------------------------------Hinweise---------------------------------
# Bitte ueberpruefen Sie, ob Sie die richtige Ausgabe eingestellt haben.
# Vorheriger Zustand, der den Test besteht:                Hauptstrasse Rot Gelb
# Ueberpruefen Sie ab besteheden Zustand, ob Sie:
# - Das richtige load Signal eingestellt haben.
# - Den richtige init Wert eingestellt haben.
# - Ihr Programm korrekt auf dem naechsten Zustand eingestellt wurde.
```

Listing 5: Signal mismatch error

This error occurs when the testbench finds an output that does not match the desired output. In this case, the signal **hs_f_rt** is 0, which is different from the expected output 1. There are many reasons for this error. First of all, you should check in the error state (**Hauptstraße Grün** in this example) to see if you programmed the output correctly. If you are sure that you have correctly programmed the desired output, start from the nearest state where your design is correct (**Hauptstraße Rot Gelb** in this case) and check the signals **init** and **load** of all states to the fault state to see if there exists any mistakes. If you cannot find the problem, then check state transitions and the conditions to change the states in your module, such as **ready**, **hs_f_an** or **ns_f_an**...If the next state is wrong, the output of your design will be obviously wrong.

The hints provided by the testbench do not cover all the reasons for the error. After trying all of them, you can discuss with the tutors for further support.

3 Signal waveform with EPWave

Observing the output waveform is an extremely effective way to verify the design. EDA provides EPWave tools to help users observe the signal waveform in the design. To use EPWave, please check the box "**Open EPWave after run if not**" as in the figure 1.

Figure 1: EPWave setup

After run the testbench, the EPWave window will automatically pop up. Click on **Get Signals** to choose which signals you want to verify as in the figure 2.

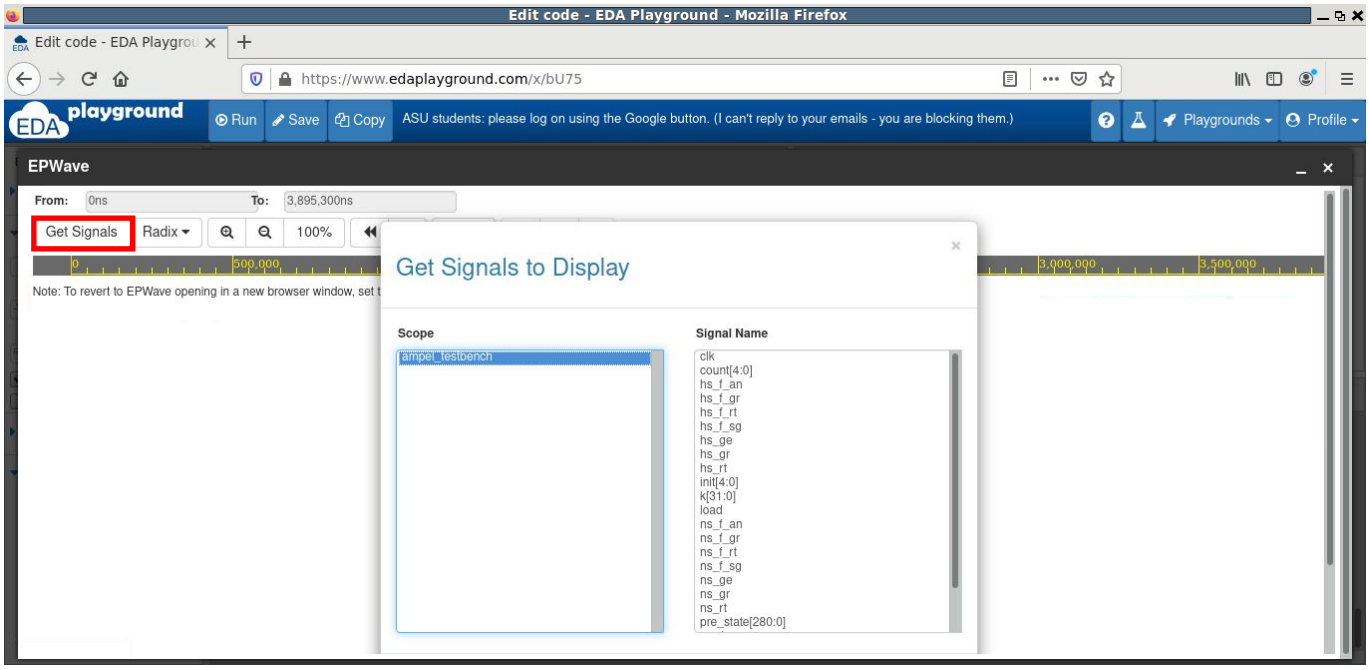

Figure 2: EPWave signal selection

After selecting the signals, we can observe the signals as in the figure 3.

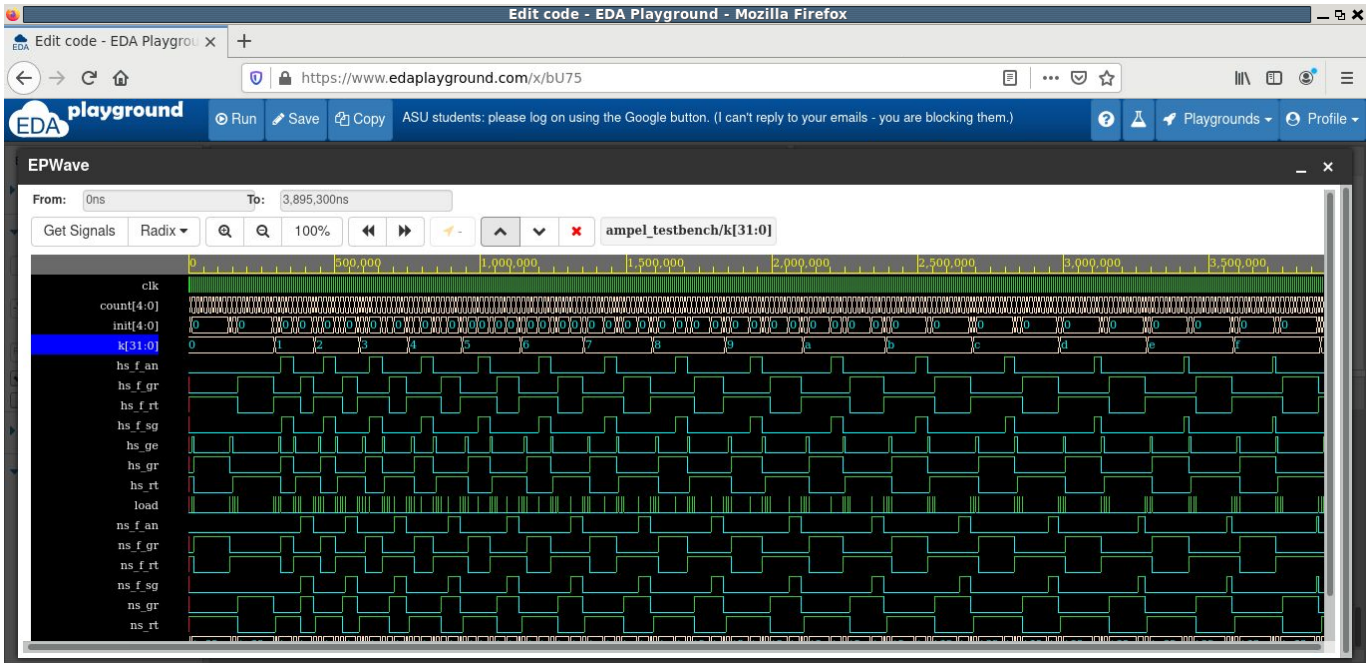

Figure 3: EPWave signal waveform

4 Design synthesis

4.1 Mentor Precision 2019.2 setup

After the design has passed all the tests. Design needs to be tested to see if the design can be synthesized on hardware. There are a number of ways to write code and functions that can be simulated but cannot be synthesized to the hardware. The code needs to be altered or removed from the design in order to be implemented in hardware.

EDA provides Mentor Precision 2019.2 to help users check if their design can be synthesized on hardware. To use the tool, first choose Mentor Precision 2019.2 as the Tool & Simulation. Then create a run.do file as in figure 4. Finally, click run to see the synthesis process. When the design is synthesizable, you will get **"Done"** without any error message as the result as in figure 4

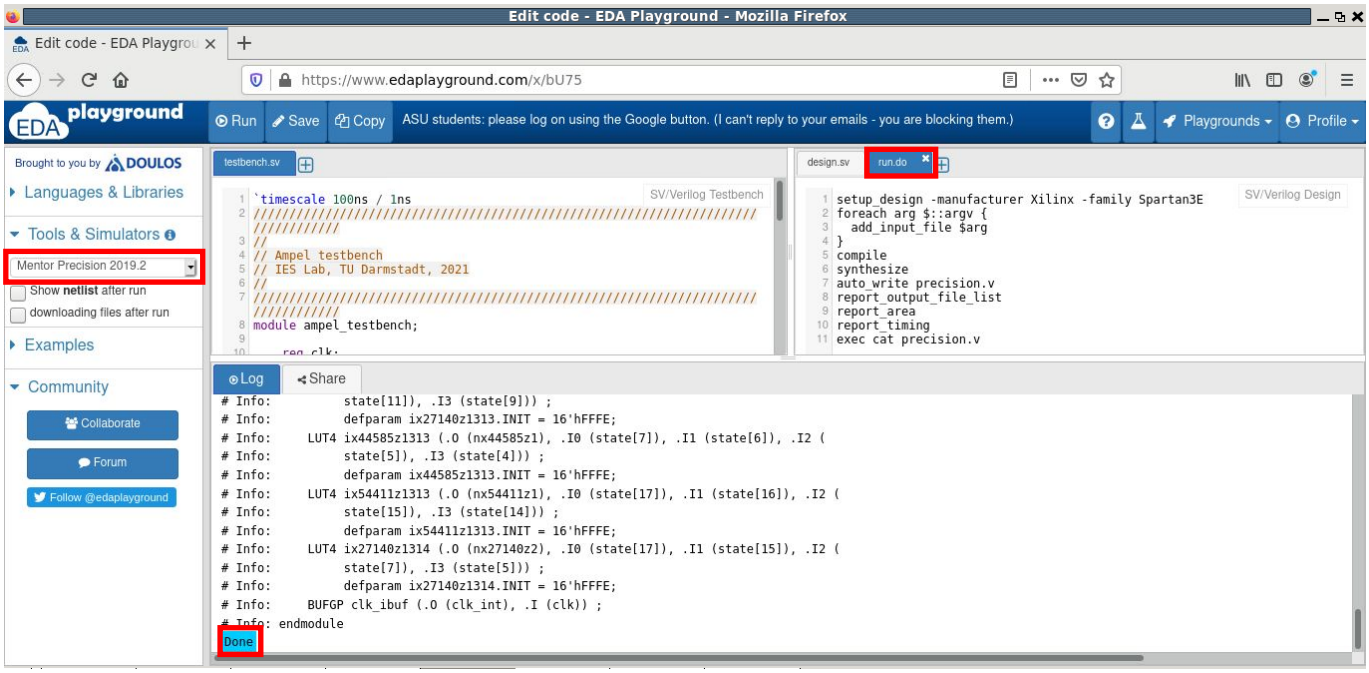

Figure 4: Mentor Precision 2019.2 setup

```
setup_design -manufacturer Xilinx -family Spartan3E
foreach arg $::argv {
  add_input_file $arg
}
compile
synthesize
auto_write precision.v
report_output_file_list
report_area
report_timing
exec cat precision.v
```

Listing 6: run.do

4.2 Mixed blocking and non-blocking assignments in one always block

A common mistake is the designer mix blocking and non-blocking assignments in one always block. In case, this error occurs, you will get an error message as in figure 5

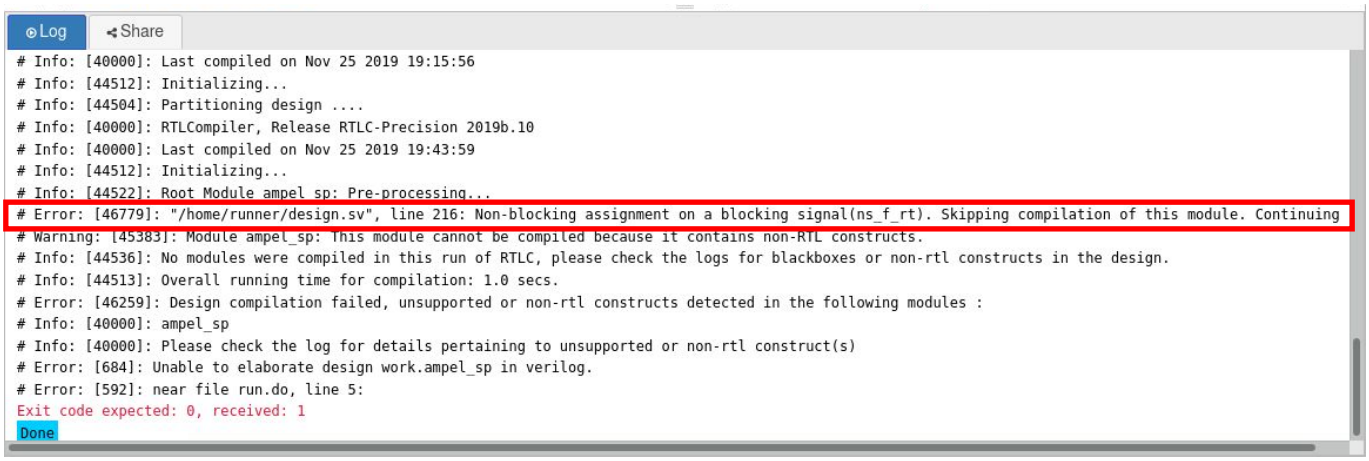

Figure 5: Mixed blocking and non-blocking assignments in one always block

4.3 Connection of multiple drivers to one signal

Another common mistake is "**Connection of multiple drivers to one signal**". The cause of this error is because there is a signal in your design that could be changed in the same clock (maybe a register is being assigned in two different always blocks). You should rewrite the code in such a way that you are sure this will not happen. In case, this error occurs, you will get an error message as in figure 5

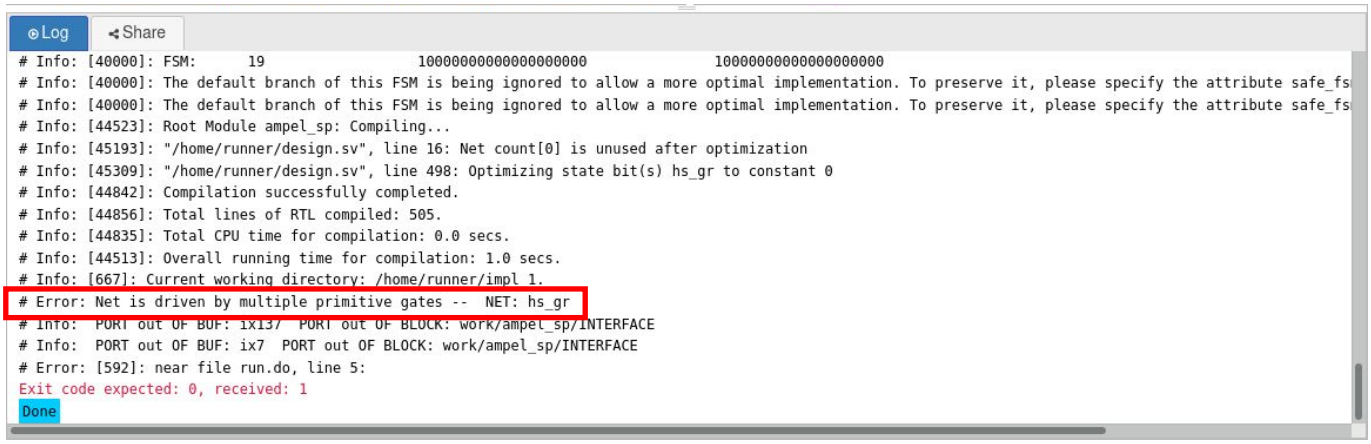

Figure 6: Connection of multiple drivers to one signal