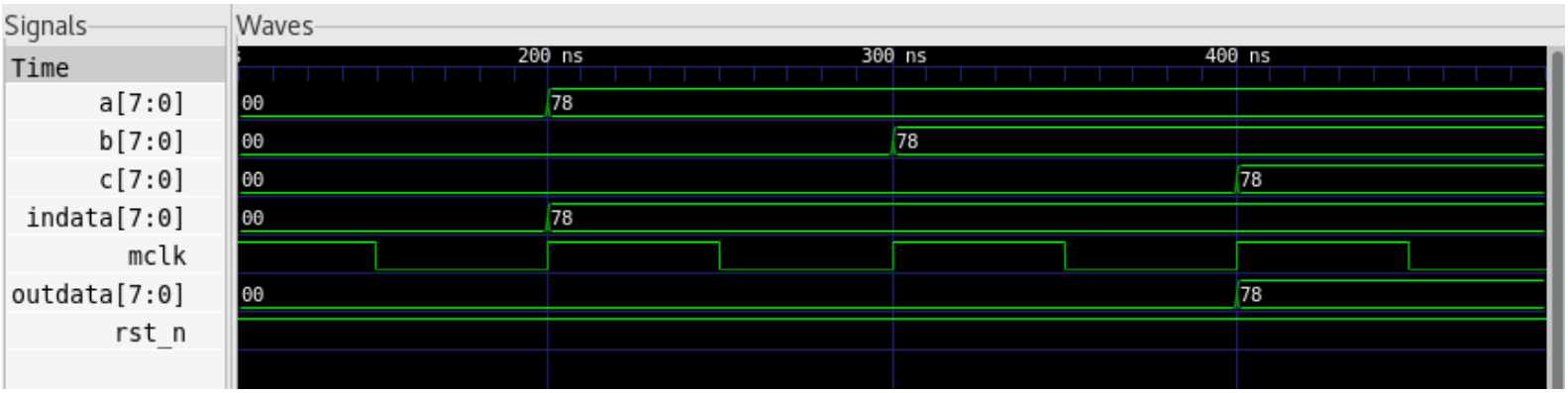
a)

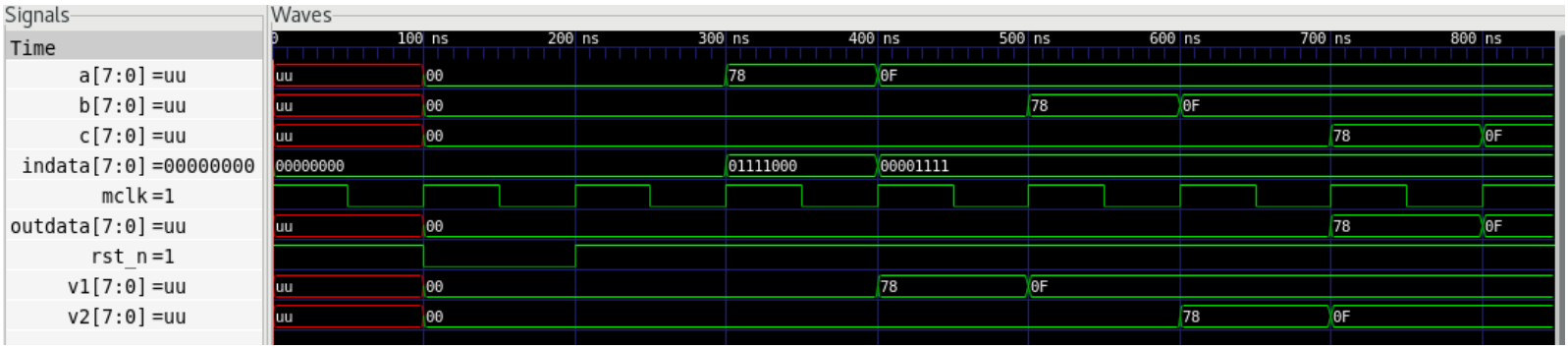
When **rst_n** ='1' and **mclk** is on the rising edge signals are allowed to change from 0 to something else. **a** gets the value of **inndata** on the rising edge of **mclk**, but this value is not set until the end of the process, so all other signals are still 0 on the clock cycle where **a** is set to '1'. On the next rising edge of the clock, the variable **v1** is assigned '1' and **b** is assigned the value of **v1**, all the other variables retain their value because **b** is not set before the end of the process. On the rising edge after **b** is updated to '1', **v2** variable is assigned '1' and thus **c**-signal is updated to '1'. Because **outdata** is updated outside the process, this is updated on the same clock cycle as the **c**-signal.

The reason for the delay, is that the signals (a, b, c) have to propagate the signal before it reaches **outdata**. It is not possible to update a signal several times per clock cycle, that is why it takes two clock cycles for **outdata** to update.



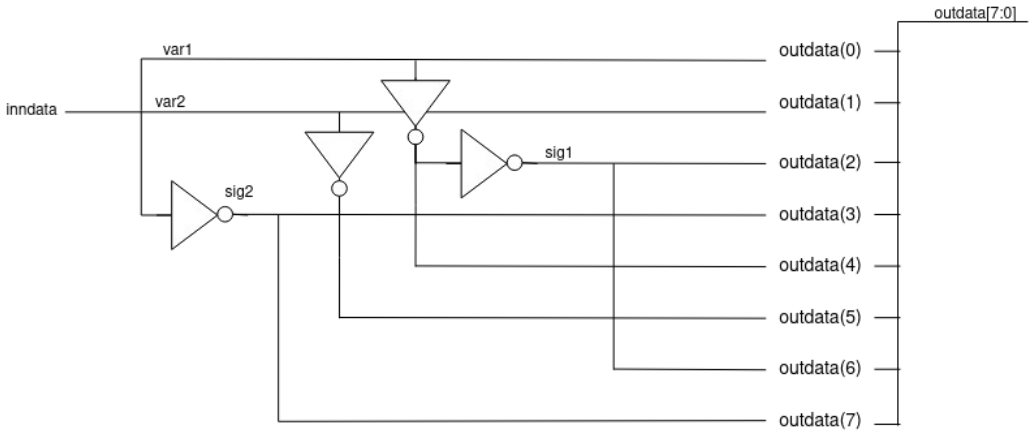
b)

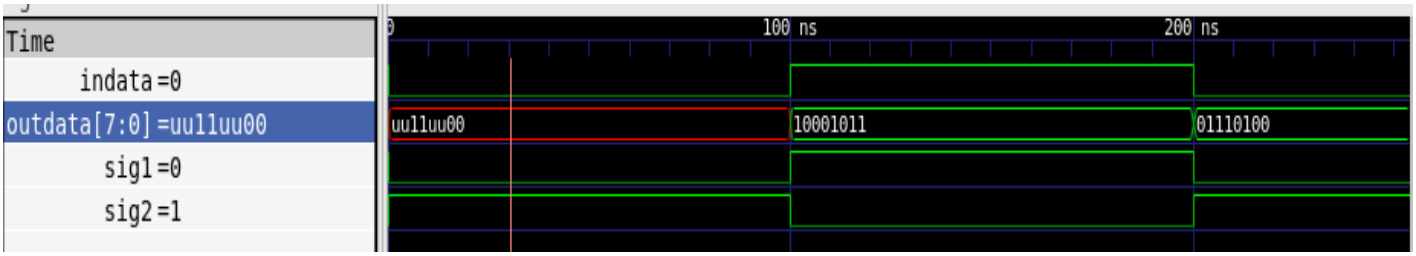
1) Now the **outdata** signal change four clock cycles after the **inndata** is updated as opposed to two clock cycles when **v1** and **v2** were variables.
2) When **v1** and **v2** were variables they could send the byte to **b** and **c** on the same clock cycle as they "received" it. But when they were changed to signals, **b** and **c** had to wait one clock cycle to be updated.
3) That is because it gets its value from the other signals, and they do not get initialized until later.



c)

1) **sig1** and **sig2** are signals, therefore the assignment of new values to these signals does not come into effect until after the process is done. **output(7 downto 6)** and **output(3 downto 2)** both get their values from **sig1** and **sig2** that is why they are always the same.
2) **var1** and **var2** are variables. **output(1 downto 0)** gets values from these two variables, after that the variables are given new values before **output(5 downto 4)** gets values from these variables. Since **var1** and **var2** get updated before **output(5 downto 4)** is set, it is different from **output(1 downto 0)**.
3)





d) This is because **sig1** and **sig2** cannot propagate a signal until they are set. And that can only be done the next time the process is triggered by the **indata**. Because the two signals are no longer in the sensitivity list, there is a delay of updating outdata.