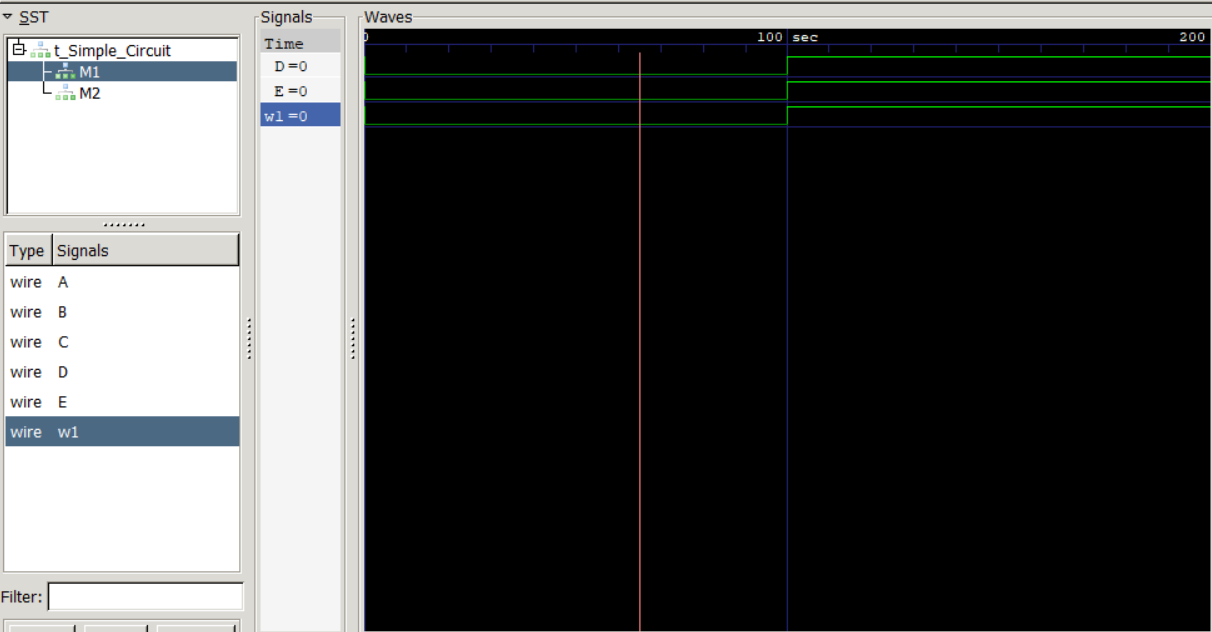
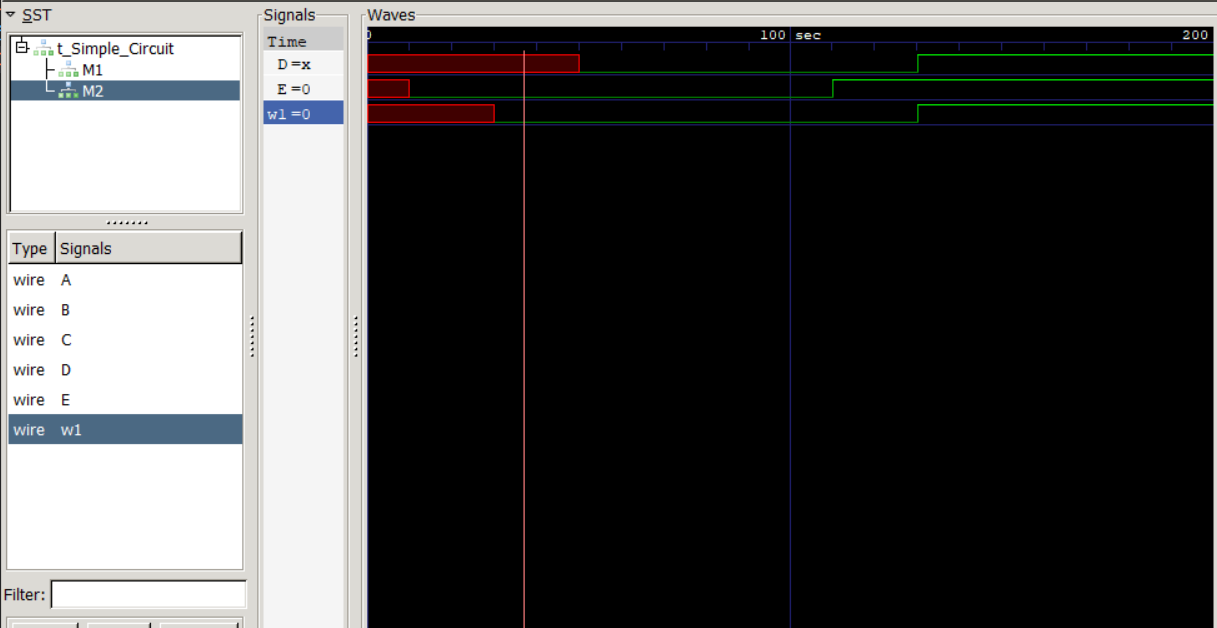
(1)

*Simple\_Circuit.v*



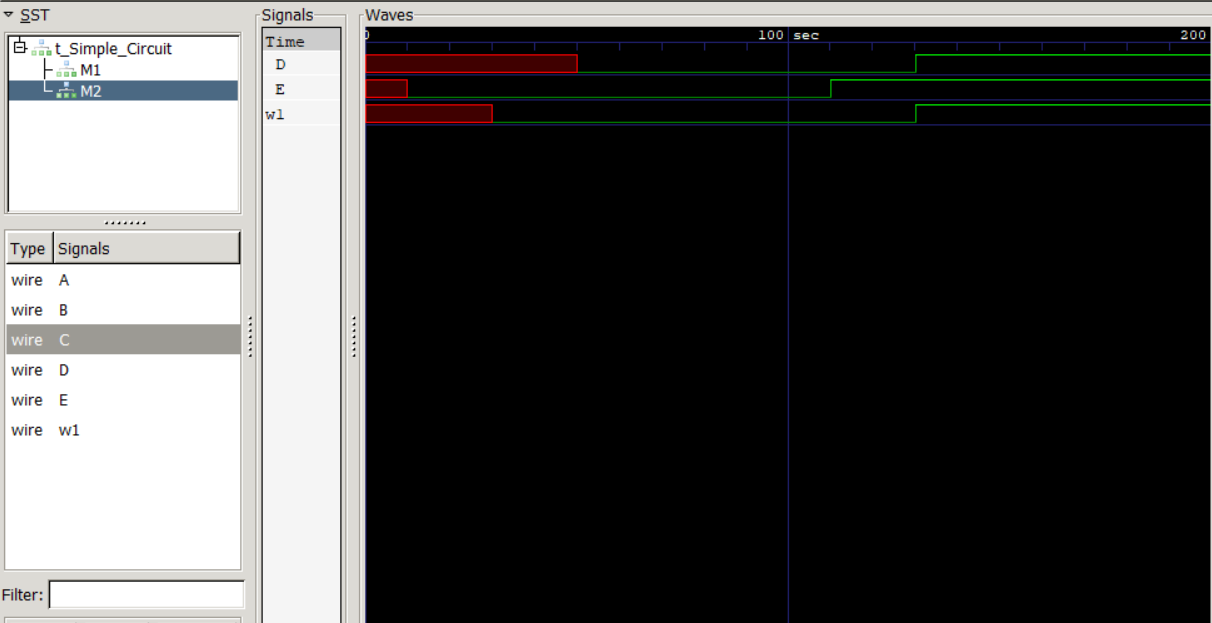
*Simple\_prop\_delay.v*



**Explanation:**

The propagation delay of AND gate is 30 time units, and thus in *Simple\_prop\_delay.v,* the wave of w1 translates 30 time units to the right. Also, the wave of E translates 10 time units to the right, i.e. E=1 after 10 time units. As a result, after 20 more time units, D becomes 1 although w1=1 just propagates to the OR gate. This is because an OR gate outputs 1 once one of its inputs is 1. Therefore, the wave of D translates 10+20=30 time units to the right.

(2)



**Explanation:**

The waveforms are identical, for in synthesizable module, all statements are evaluated concurrently. Thus, interchange of statements won’t arise difference.

(3)

一張含有 文字, 室內, 監視器 的圖片

自動產生的描述一張含有 文字, 監視器, 室內, 時鐘 的圖片

自動產生的描述

**Explanation:**

From the circuit diagram, we can derive its boolean expression:

(A+B’)C + (BC+D’)A

Which can be further reduced to:

C(A+B’) + AD

To evaluate the correctness of the modules, I will derive the truth table of the above boolean expression first.

**Truth table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | F |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

As we can see, the value of F in the truth table can accurately correspond to the waveform. Furthermore, the waveforms of all three modules are identical. Hence all three modules are correct.

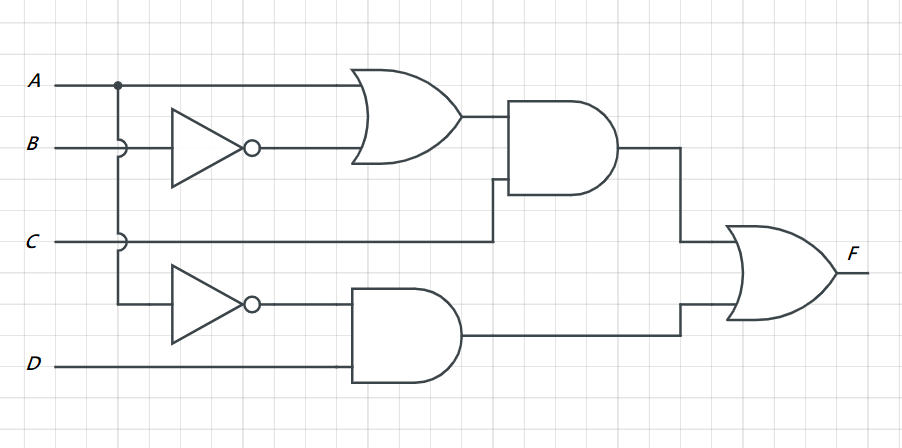
(4)

No, it isn’t.

The boolean expression of the original implementation is:

(A+B’)C + (BC+D’)A, GIC = 14

* AC + B’C + ABC + AD’
* AC + B’C + AD’
* **C(A+B’) + A’D, GIC = 10** (simplified expression)

****

(5)

**Problems & Difficulties:**

Knowing the concept of logic gates and circuits is different from implement it by HDL: I have to get familiar with the syntaxes of Verilog. Although it slightly resembles other programming languages, like C, but its character of describing hardware distinguishes itself, which makes learning a challenge.

By the way, choosing a suitable tool to draw a circuit diagram is also important, for it can save a lot of time!