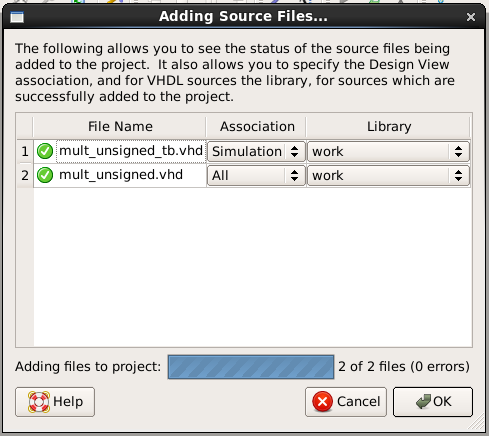
**Computer Logic – Practical 3**

**Objective:**

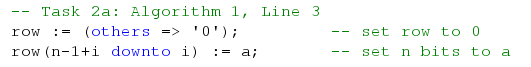
To investigate unsigned and signed multipliers.

**Tasks:**

1. A new project called *“Project3”* was created and the provided files *“mult\_unsigned.vhd”* and *“mult\_unsigned\_tb.vhd”* were added to it from *Project: Add Source*. The association was set to *All* for the *mult\_unsigned.vhd* file and to *Simulation* for the *mult\_unsigned\_tb.vhd* file.



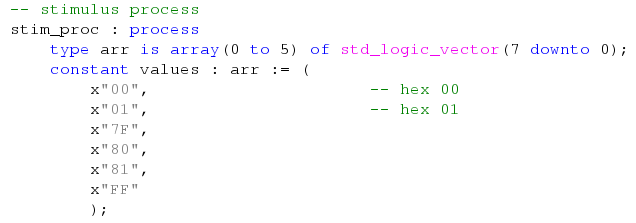
1. The file *mult\_unsigned.vhd* was then opened. This included a *shift\_mux* process which contained for loops and also if statements. However, this process was not complete and so was modified as follows:
2. The contents of row corresponding to Algorithm 1, Line 3 were generalized to be true for all iterations by altering the code as shown.



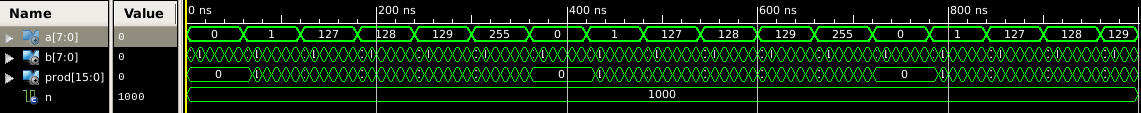
1. Algorithm 1, Line 4 was modified to add the value of the row to the total in the if statement as can be seen.



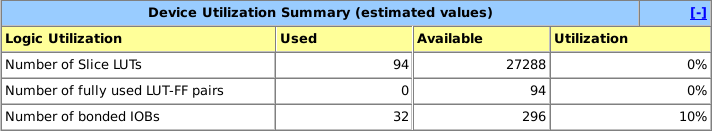
1. The test bench *mult\_unsigned\_tb.vhd* was then opened and modified with an additional 4 hexadecimal values so that there are 6 in all.



After this was done, the test bench was simulated using *Behavioral Simulation* to test the multiplier as can be seen in the next snippet. The Radix was changed to *Unsigned Decimal* since the multiplication was unsigned and all 36 possible products were verified to give the expected output for correctness sake.



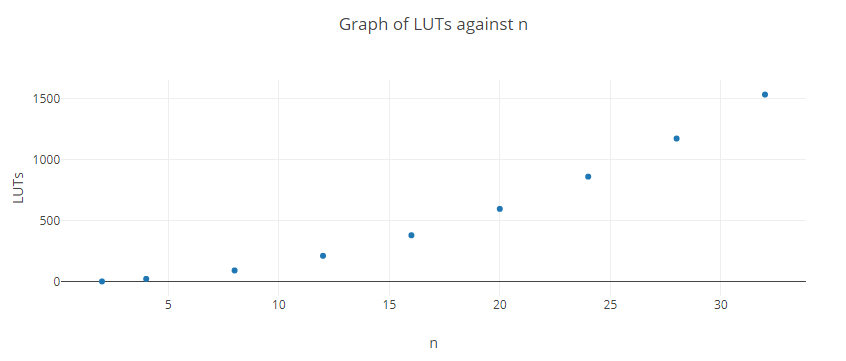
1. The hardware area of the multiplier was measured next. This was done automatically by the program when double clicking the *Synthesize* option found in the *Implementation view*, on the *Processes* pane when the multiplier is selected. The hardware footprint of the design was then found from the *Design Summary*, in the *Device Utilization Summary* when reading the *Number of Slice LUTs*. For this particular multiplier, the hardware area was found to be 94.

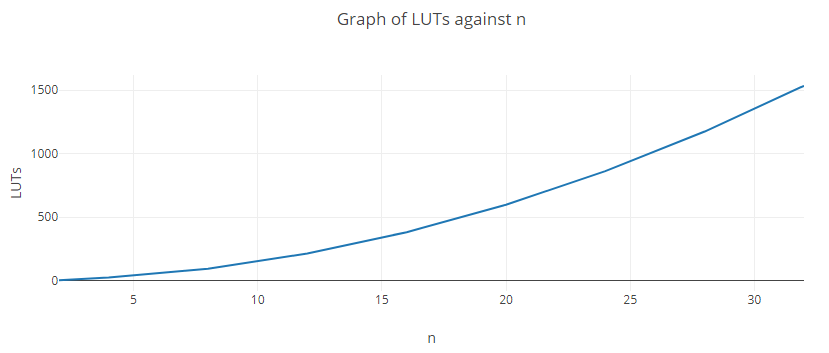


1. In this next step, *n* was changed in the range of 2 ≤ *n* ≤ 32 inside the generic part of the entity and step 4 was repeated each time to obtain a measure of the hardware area for these different values. The results are listed on the table below

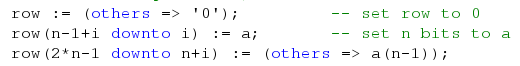
|  |  |
| --- | --- |
| **n** | **LUTs** |
| 2 | 4 |
| 4 | 25 |
| 8 | 94 |
| 12 | 214 |
| 16 | 382 |
| 20 | 598 |
| 24 | 862 |
| 28 | 1174 |
| 32 | 1534 |

1. A graph of LUTs (y-axis) against n (x-axis) was plotted and this graph seemed to relate mostly to a quadratic curve. The reason why is simple. The larger the value of *n*, the bigger the hardware area required so it would not make sense for such a graph to represent a square curve or a cubic one implying that for certain small values of *n*, the hardware area required is larger than for other values of *n* in which *n* is larger than before. The 2 snippets below represent this graph distribution, one with the plotted points and the other with the points connected.



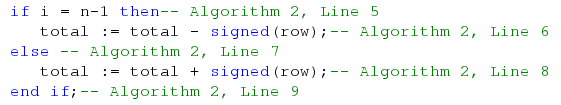


1. The whole *mult\_unsigned.vhd* file was copied to a new project file entitled *mult­\_signed.vhd*. In this file any appearances of *mult\_unsigned* were changed to *mult\_signed*. The *shift\_mux* process was also altered to perform signed multiplication instead of unsigned as explained in the following 3 steps:
2. All appearances of *mult\_unsigned* were changed to *mult\_signed*.
3. *a* was sign-extended when setting the row variable this time. This was done by adding an extra line as shown together with the 2 previously written lines.



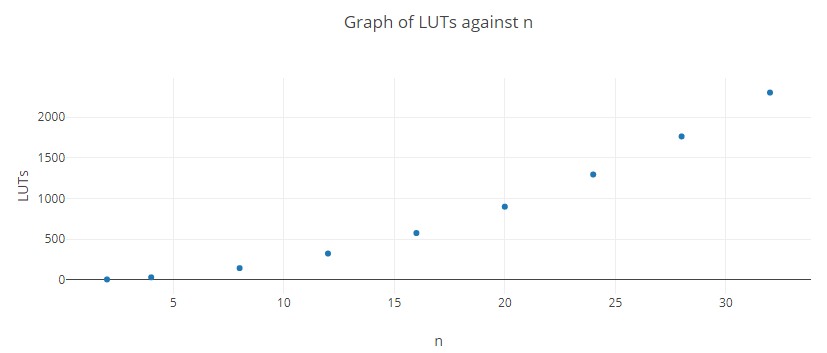
This line copies the MSB of *a* and adds it to the sign extension bits

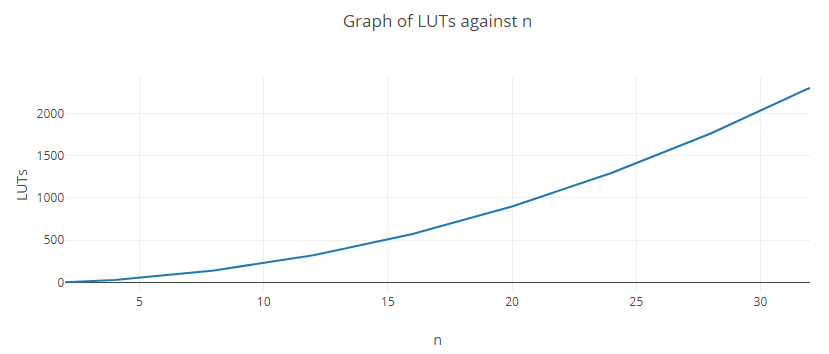
1. The line *total := total + unsigned(row);* was replaced with the following conditional statement since in signed multiplication, both addition and subtraction are performed when calculating the total.



1. The file *mult\_unsigned\_tb.vhd* was copied to a newly created file *mult\_signed\_tb.vhd* which was linked to this project. All appearances of *mult\_unsigned* were once again changed to *mult\_signed* and the test bench was simulated and checked to be correct. The radix of the simulation was changed to *Signed Decimal*.
2. The file *mult\_signed.vhd* was set as the **Top Module** and a measure of the hardware area in the range 2 ≤ *n* ≤ 32 was obtained. A table with the possible values was written down. From this table, 2 versions of a graph of LUTs (y-axis) against n (x-axis) were drawn up as shown. The first version shows the plotted points and the second shows the points connected. This graph, just like the one shown in step 6 is a quadratic curve for the same reason explained already.

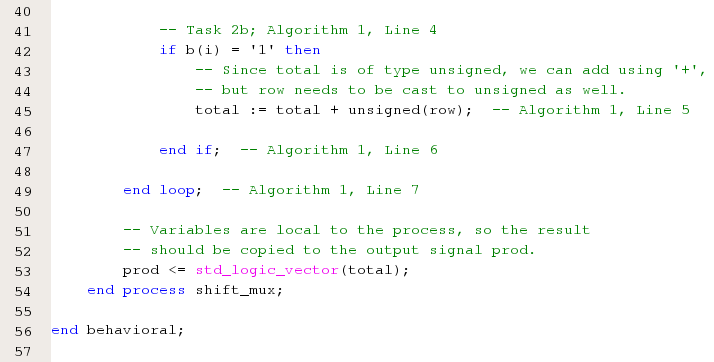
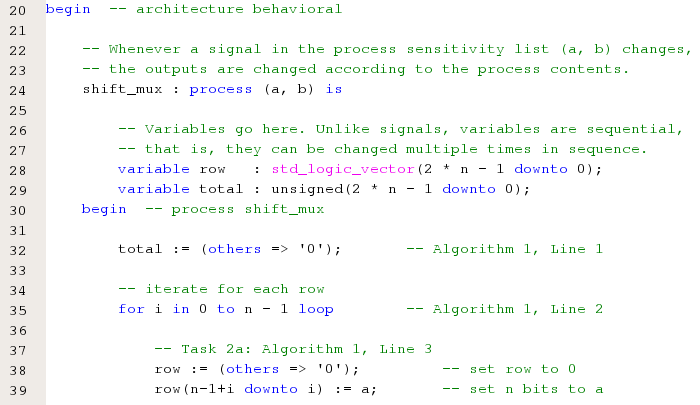
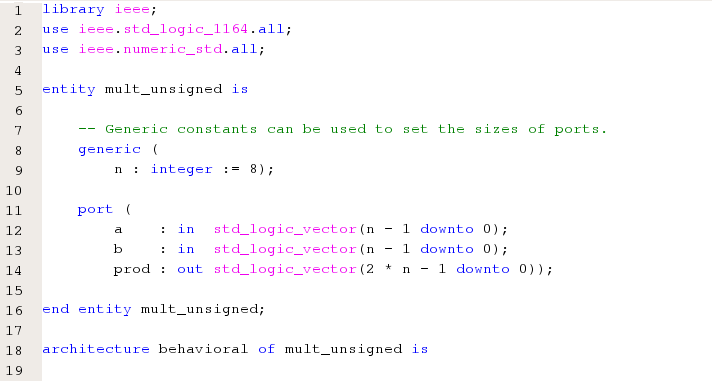
|  |  |
| --- | --- |
| **n** | **LUTs** |
| 2 | 4 |
| 4 | 30 |
| 8 | 143 |
| 12 | 323 |
| 16 | 575 |
| 20 | 899 |
| 24 | 1295 |
| 28 | 1763 |
| 32 | 2303 |



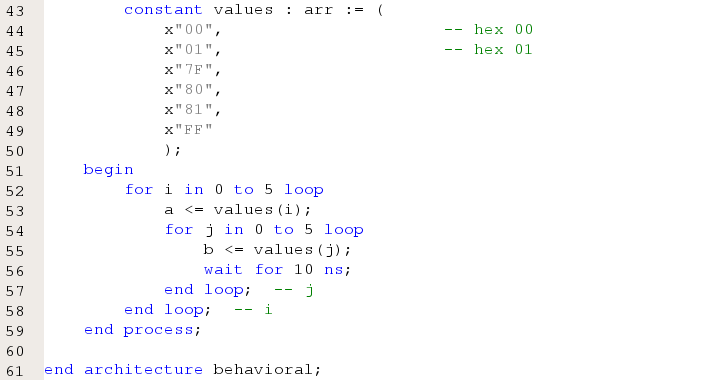
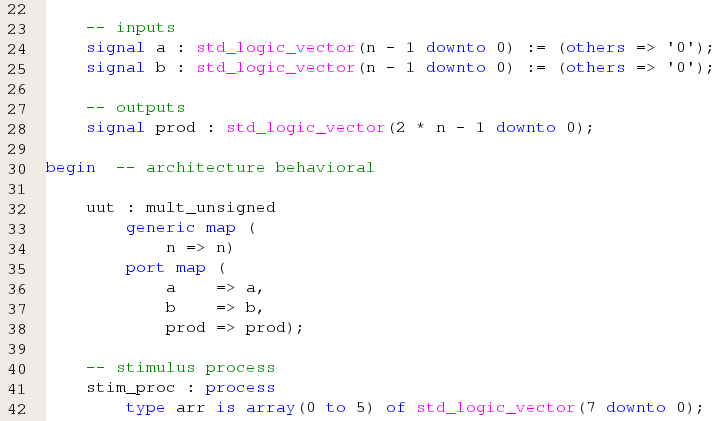
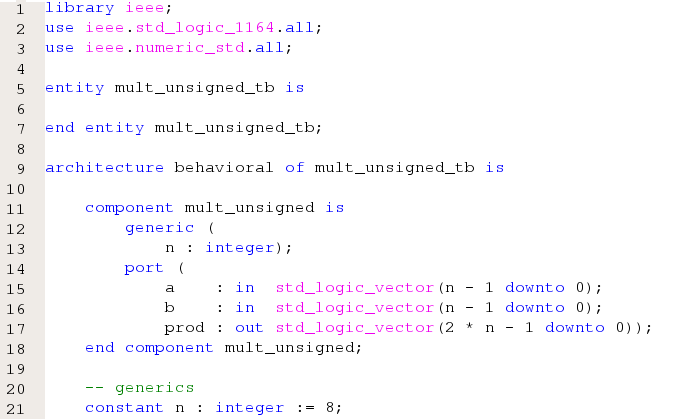


**Appendix (Code):**

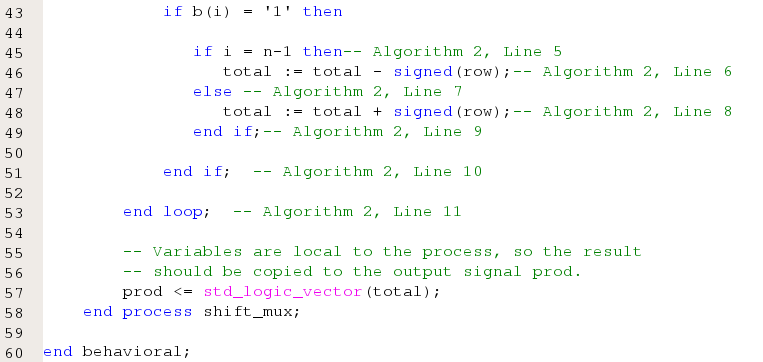
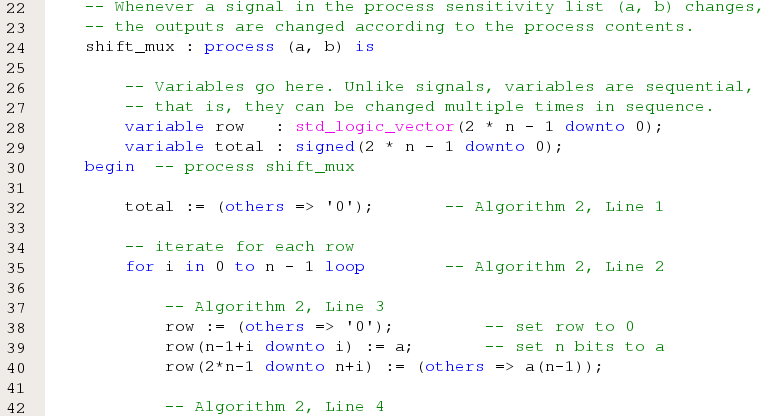
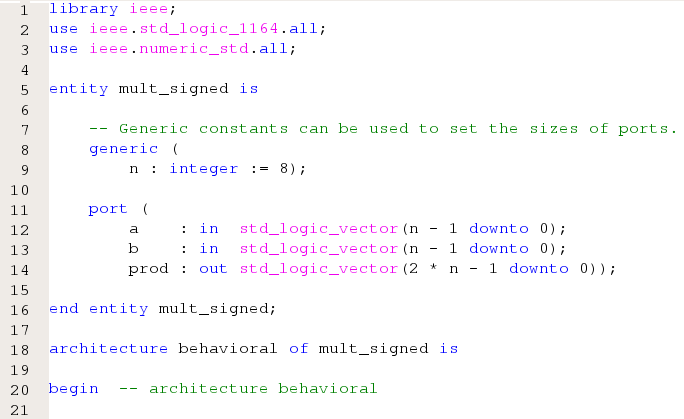
***mult\_usnigned.vhd:***



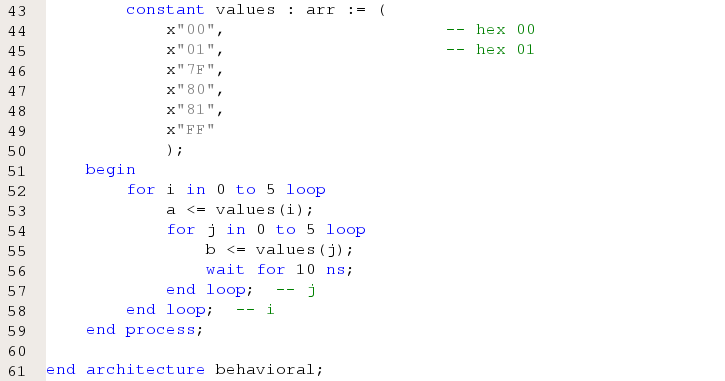
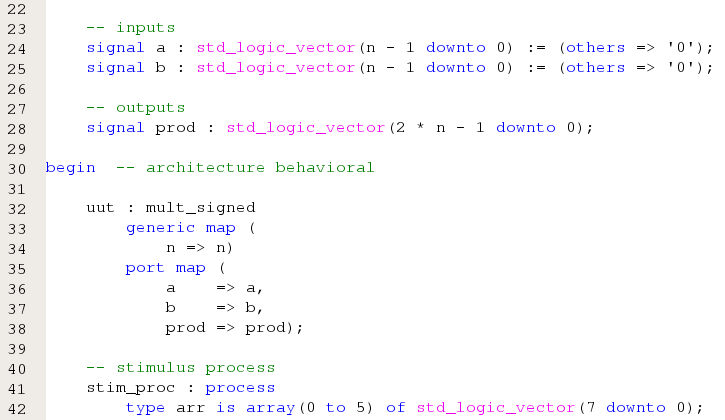
***mult\_unsigned\_tb.vhd:***



***mult\_signed.vhd:***



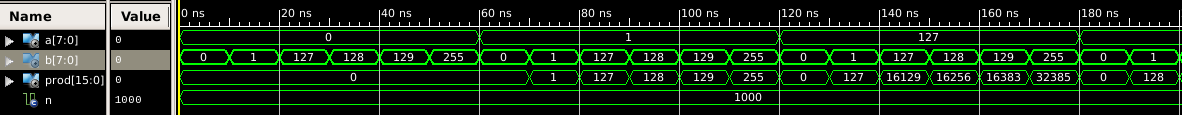
***mult\_signed\_tb.vhd:***

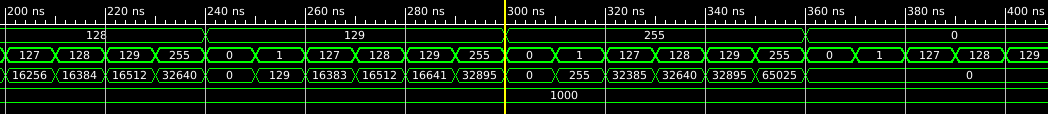


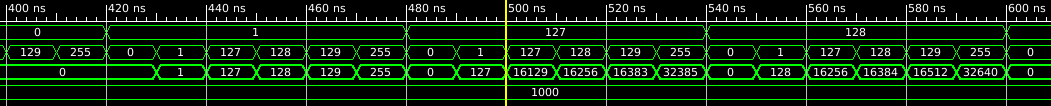
***Simulated Behavioral Models:***

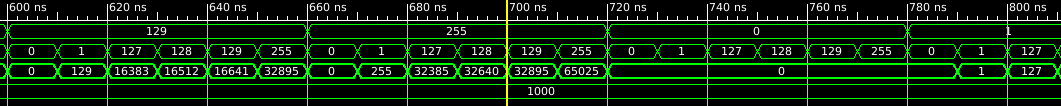
Each snippet below shows only 200ns of the Simulated Model. One snippet leads onto the next 200ns.

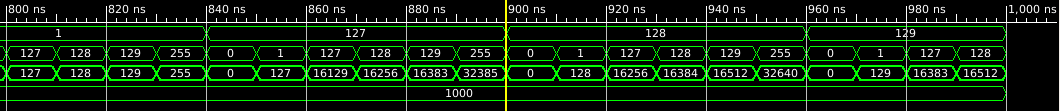
***mult\_unsigned\_tb.vhd:***



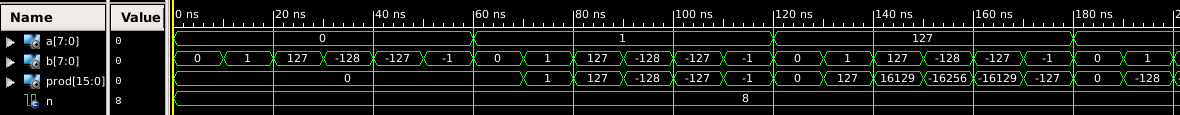


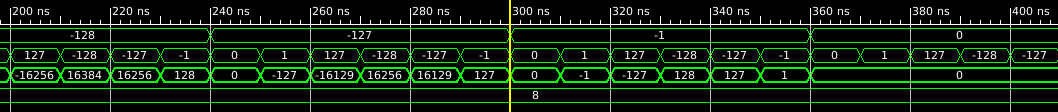


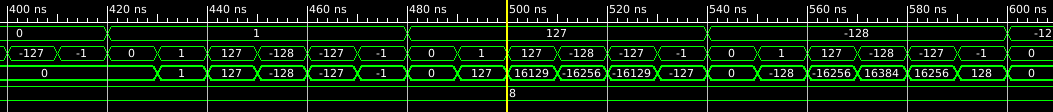


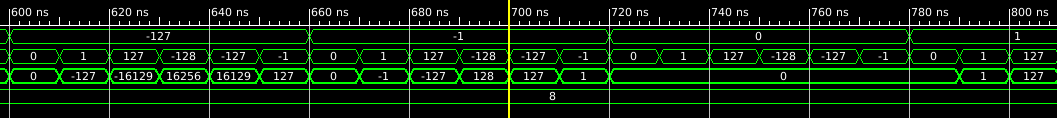


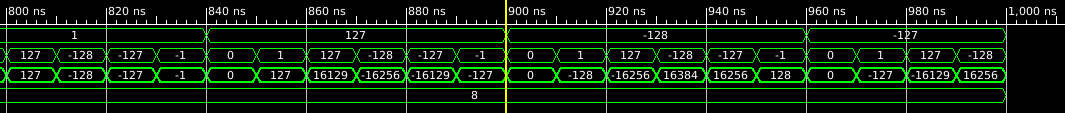
***mult\_signed\_tb.vhd:***











**Conclusion:**

Signed and unsigned multipliers were successfully implemented in this practical.