

JK LAKSHMIPAT UNIVERSITY

DIGITAL CIRCUIT AND SYSTEMS  
(EE1120)

Activity 09

2 – Bit Multiplier using VHDL language.

Date : 14th March 2024

Name : Bobby Sharma

Roll no. = 2023BTECH023

# AIM: Design and Simulation of 2bit multiplier (Structural Modelling) using VHDL language using Xilinx ISE Tool.

SOFTWARE REQUIRED: Xilinx ISE tool in your device.

THEORY: A 2-bit multiplier is a digital circuit designed to perform multiplication on two binary numbers, each represented with 2 bits. The theory behind a 2-bit multiplier involves understanding how binary multiplication works and then implementing a circuit to perform this operation efficiently. Here's an overview of the theory behind a 2-bit multiplier:

* **Binary Multiplication Basics**: In binary multiplication, each bit of one number (the multiplicand) is multiplied by each bit of the other number (the multiplier), similar to the process of multiplication in decimal. The multiplication process involves shifting and adding operations. The partial products obtained from each multiplication are then added together to get the result.
* **Partial Products:** For a 2-bit multiplier, four partial products are generated, one for each combination of the bits in the multiplicand and multiplier. The partial products are generated by multiplying each bit of the multiplicand with each bit of the multiplier.
* **Addition of Partial Products**: After generating the partial products, they are added together to obtain the result. Since we're dealing with only 2-bit numbers, the final result can be at most a 4-bit number (two bits from each partial product and a possible carry bit from addition).
* **Implementation of 2-bit Multiplier:** The implementation of a 2-bit multiplier can be done using combinatorial logic circuits such as AND gates for multiplication and XOR gates for addition. The circuit consists of blocks for generating partial products and then adding them together. Various techniques such as Wallace tree multiplier or carry-save adders can be employed for efficient addition of partial products.

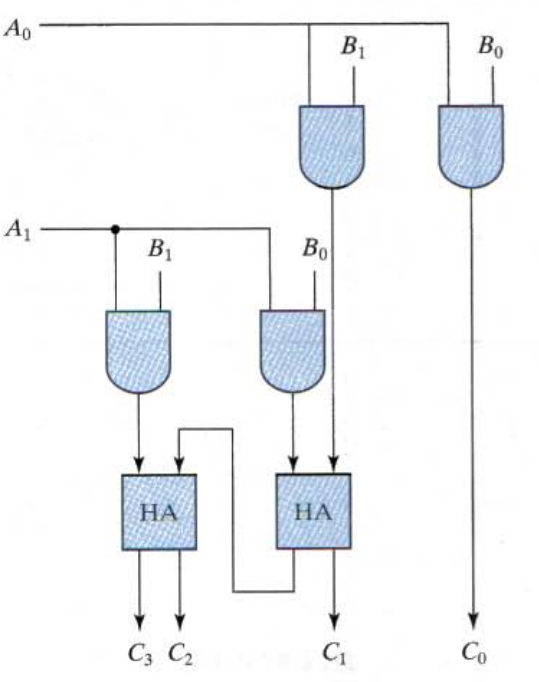


Figure 1

OBSERVATION: The observed outputs of all the basic gates are as follows:

* FULL ADDER:

VHDL Code: RTL Diagram:

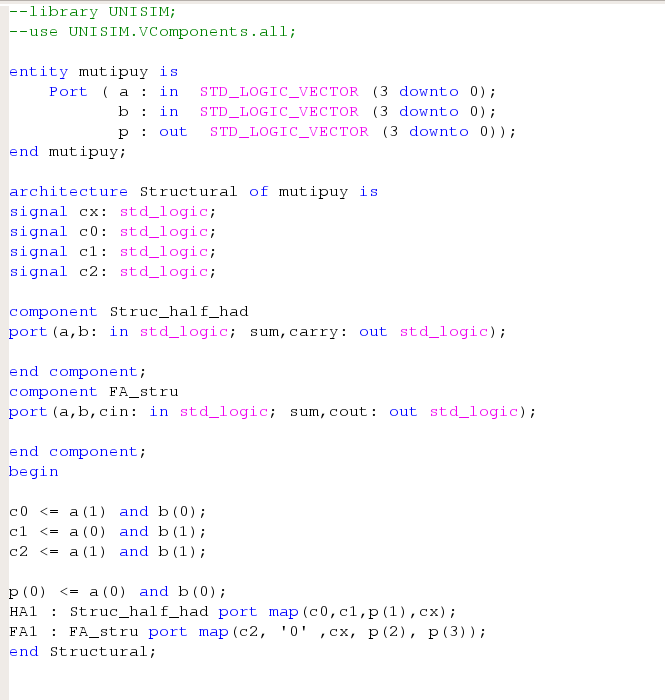
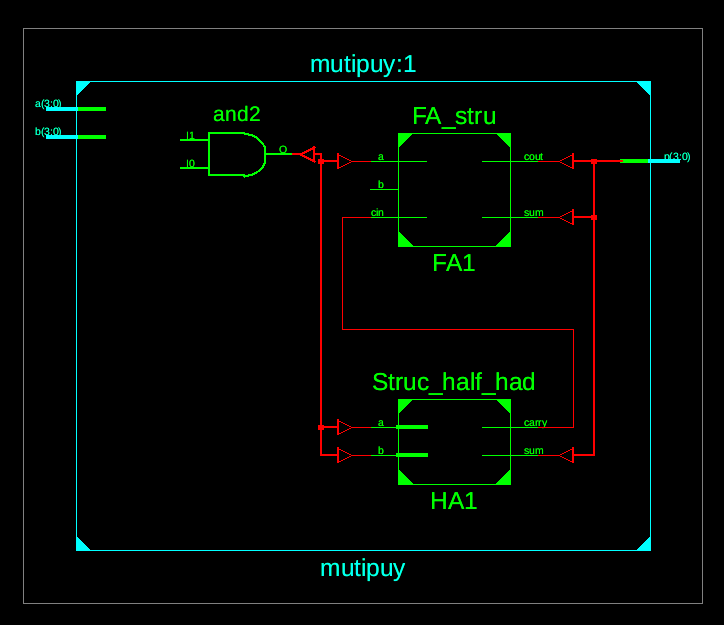
 

Figure 2 Figure 3

Test Bench Code:

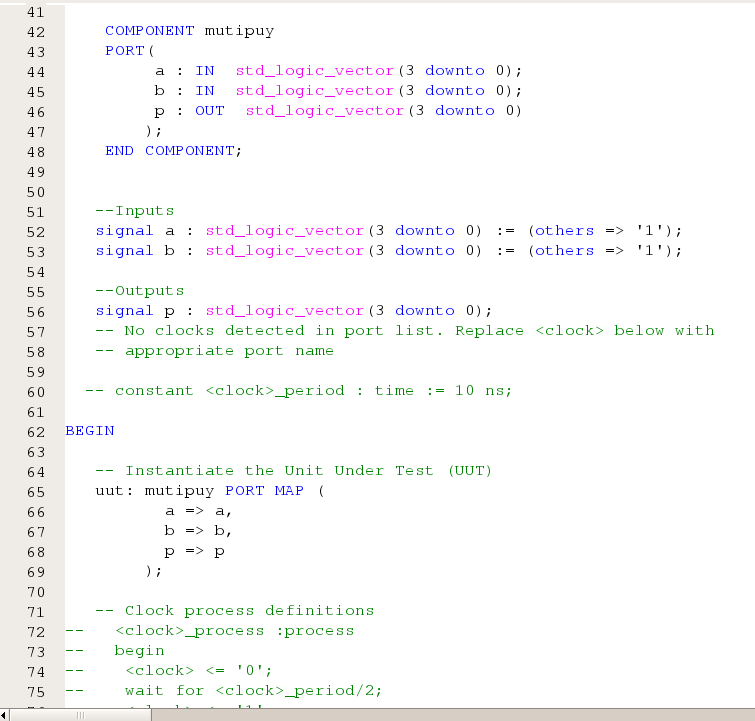


Figure 4

Waveform:

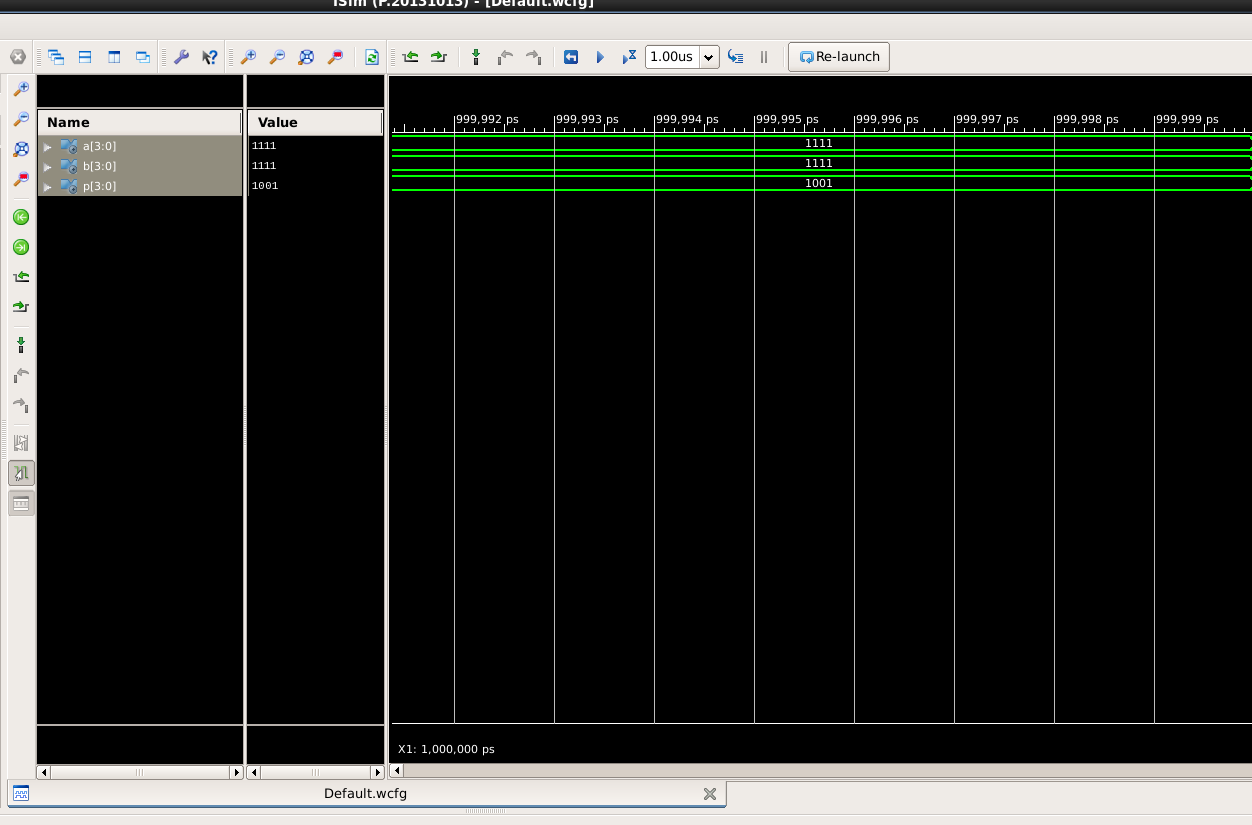


Figure 5

Here the yellow line in figure 5 represents the input (a=11, b=11) and output(sum = 1001). In which we can see the change of input signals of and b after every 100, 200 and 300 nano seconds respectively.

The truth table for a 4-bit adder lists all possible combinations of inputs (A, B, and Cin) and the corresponding outputs (Sum and Cout).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A1 | A0 | B1 | B0 | F3 | F2 | F1 | F0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

Table 1

# RESULT: We have concluded the truth table of 2-bit multiplier using VHDL language in Xilinx ISE Tool.

APPLICATION IN DAILY LIFE:

* Digital Watches and Clocks: Many digital watches and clocks use binary-coded decimal (BCD) representations for timekeeping. Multiplication operations may be required for tasks like adjusting time zones, calculating elapsed time, or performing time-related calculations, albeit with larger multipliers.
* Mobile Phones and Computers: While modern processors have significantly more complex arithmetic units, the fundamental principles of multiplication are still at play. In everyday applications, multiplication operations occur in various computational tasks, such as in gaming, image processing, financial calculations, and more.
* Financial Transactions: Banking and financial applications often involve calculations that can be broken down into multiplication operations. For instance, when calculating interest rates, loan repayments, currency conversions, or investment returns, multiplication is a fundamental arithmetic operation.
* Automotive Systems: In automotive systems, various sensors and actuators rely on microcontrollers or digital signal processors (DSPs) for control and processing tasks. These controllers often involve arithmetic operations like multiplication for tasks such as engine control, fuel injection timing, anti-lock braking systems (ABS), traction control, and more.