**Arithmetic/Logical Unit**

In this lab you will design a simple digital circuit called Arithmetic/Logical Unit (ALU) using some of the combinational building blocks. Along the way, you will use DEEDS tool to enter a schematic and simulate your design.

We need the following components:

* **2-bits Multiplexer**
* **4-bits Multiplexer**
* **Full Adder**
* **Ripple Carry Adder**

Then, you will be asked to remove the Ripple Carry Adder and to use a Carry-Lookahead Adder.

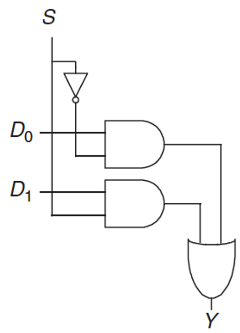
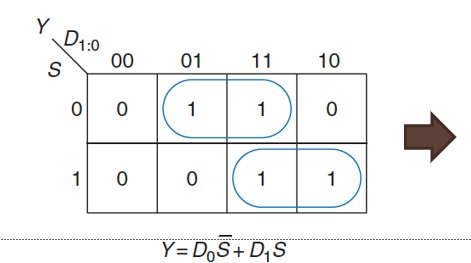
**2-bits Multiplexer**

A 2-bits multiplexer (mux) is a device that choose an output among several possible inputs, based on the value of a select signal. A 2:1 multiplexer has two data inputs and one output and a select signals.

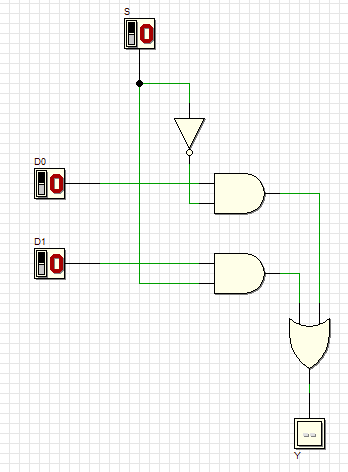
Immagine che contiene testo

Descrizione generata automaticamenteImmagine che contiene testo, orologio

Descrizione generata automaticamente Immagine che contiene tavolo

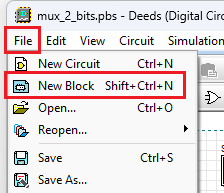
Descrizione generata automaticamente

Design the circuit using Deeds and simulate it

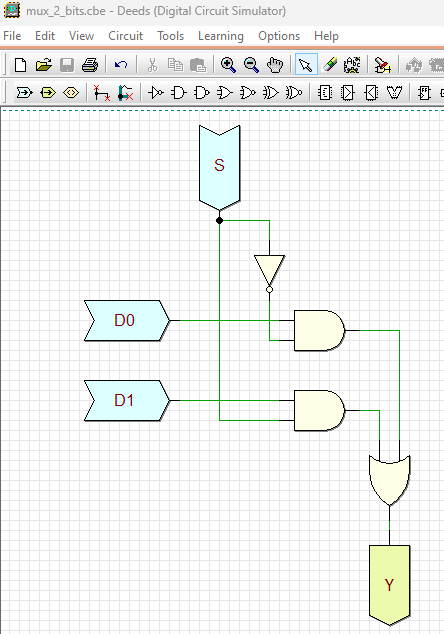


Save the file as mux\_2\_bits.pbs

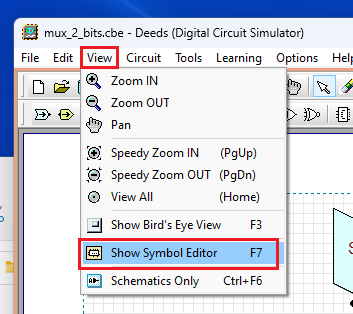
Save the logic circuit as a **block** to reuse it: Click on “File-> New Block”



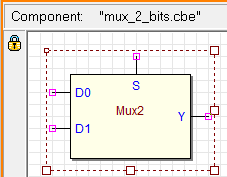
Copy the circuit into the new window and replace the input and the output with the block signal Save the block as mux\_2\_bits.cbe



To edit the block view: View-> Show Symbol Editor

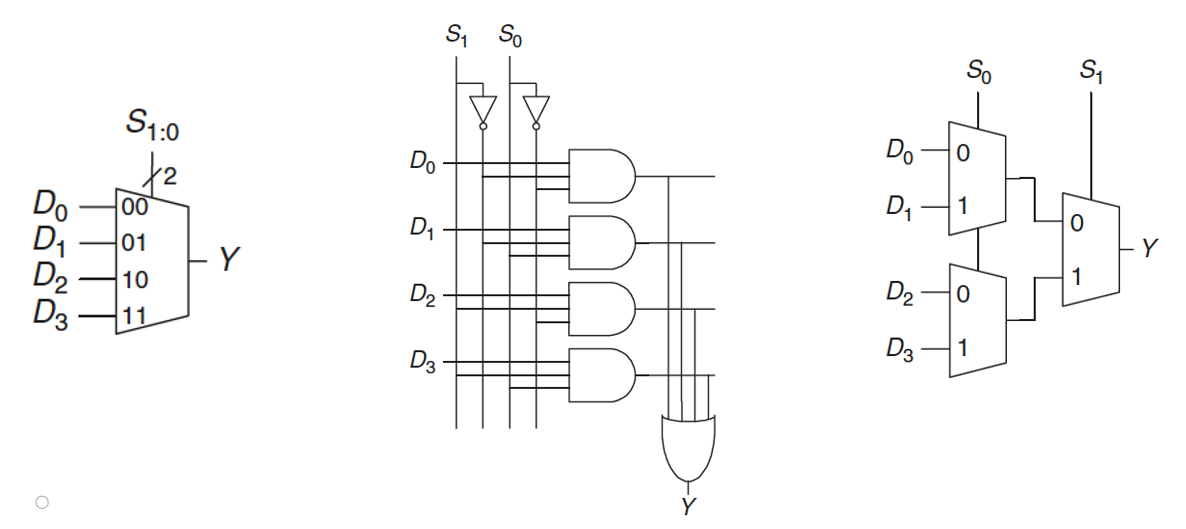


Change the view as the figure below (it may be necessary to expand the figure to move the control signal up)



**Mux 4 bits**

A 4:1 multiplexer has four data inputs and one output and two select signals are needed to choose among the four data inputs. It can be built using sum-of-product logic, or multiple 2:1 multiplexer



Create the logic circuit mux 4 bits :

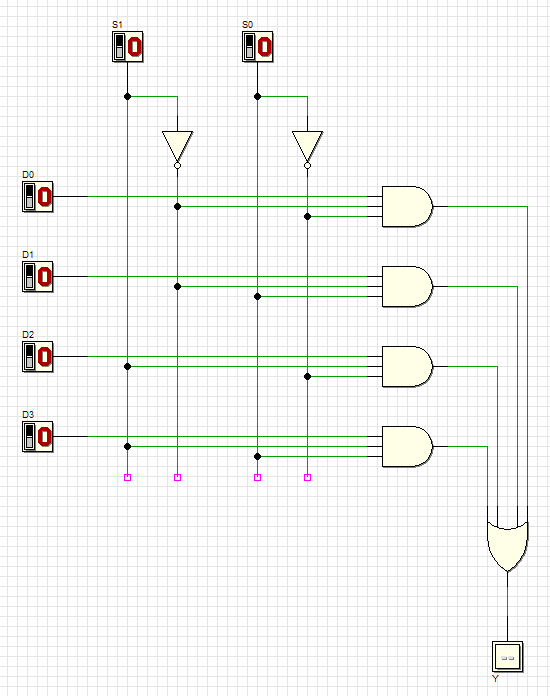
Open new circuit => “file -> new circuit”

Check that the file of the new circuit end with .pbs

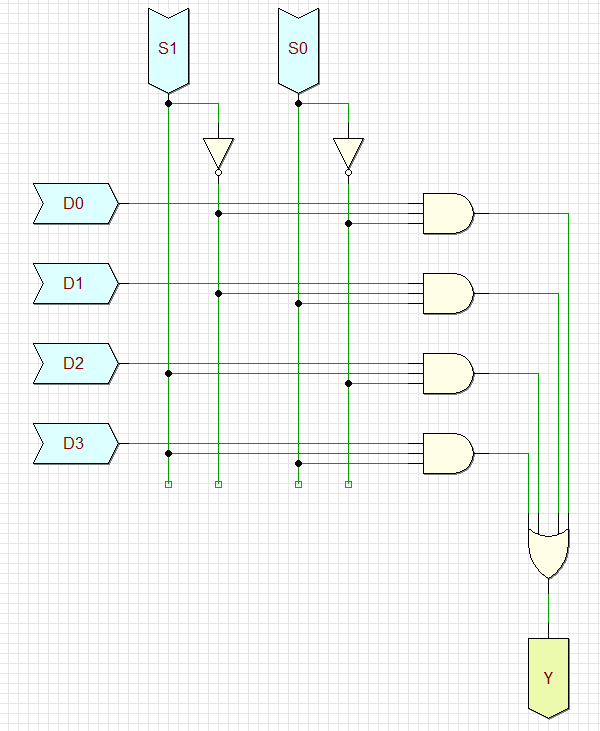
Immagine che contiene testo

Descrizione generata automaticamente

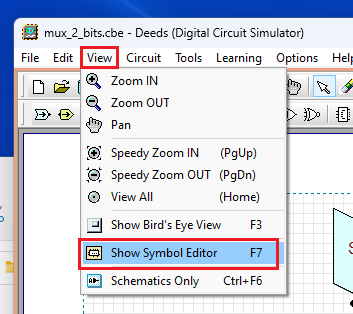
Design the full schema of the 4-bits multiplexer and simulate it



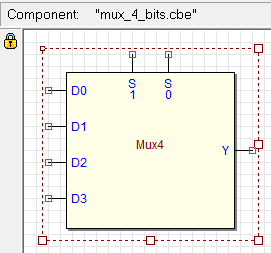
Create a block also for this component (“File->New Block”) and replace the input and the output with the block signal. Save the block as mux\_4\_bits.cbe



To edit the block view: View-> Show Symbol Editor



Change the view as the figure below (it may be necessary to expand the figure to move the control signal up)



**Full Adder**

Addition is one of the most common operations in digital systems.

Immagine che contiene tavolo

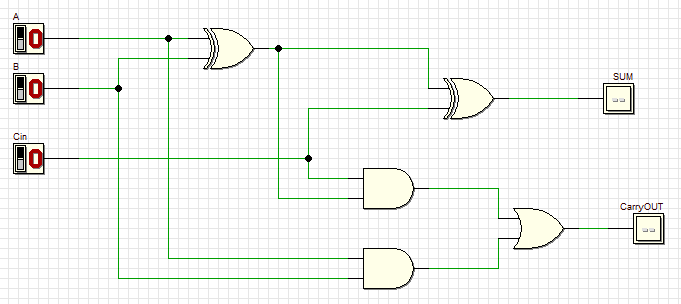
Descrizione generata automaticamente

We can optimize the Carry Out:

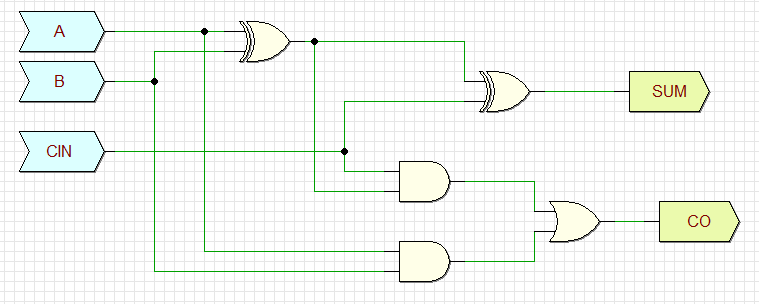
Immagine che contiene testo

Descrizione generata automaticamente

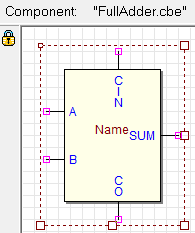
Design using DEEDS this optimized circuit of a full adder and save as FullAdder.pbs



Create the block of the Full Adder and save it as FullAdder.cbe



Change the view as the figure below



**Ripple Carry Adder**

A N-bit adder that sums two N-bit inputs and a carry in to produce an N-bit result S and a carry out. It can be implemented as a Ripple Carry Adder (Cout of one stage acts as the Cin of the next stage)

Immagine che contiene testo, orologio

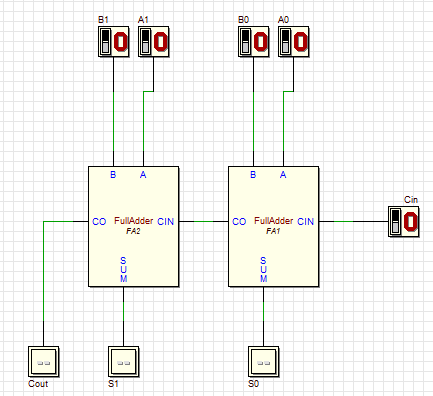
Descrizione generata automaticamente

Design a 2-bits Ripple Carry Adder. Import the 1-bit Full Adder block (FullAdder.cbe) into A new circuit using the “Custom Components”icon of DEEDS toolbar

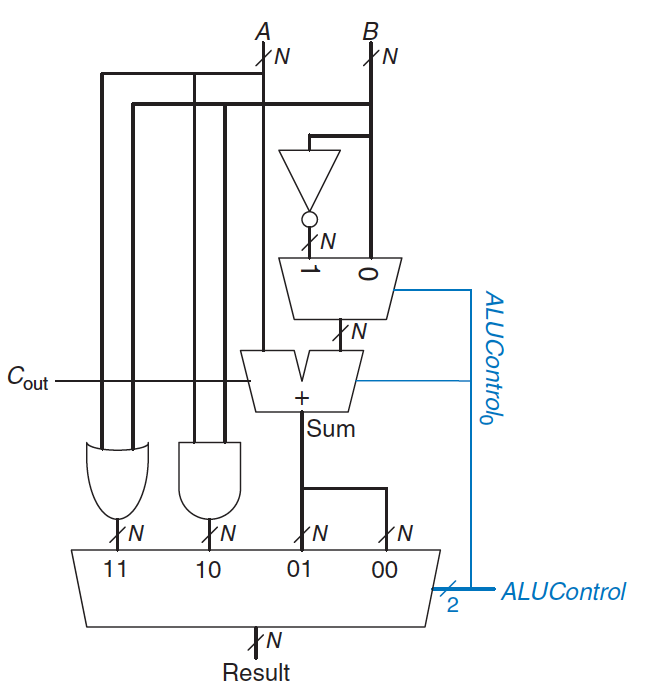
Immagine che contiene testo

Descrizione generata automaticamente

Select the file FullAdder.cbe and copy it into the circuit.

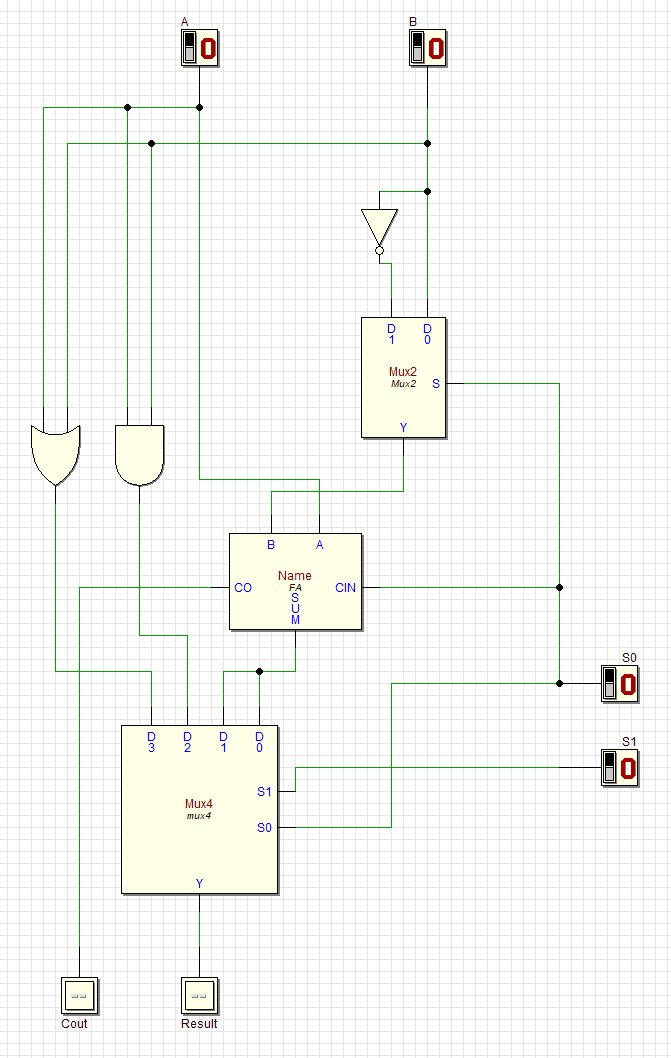


Now we can put all together to design a complete ALU

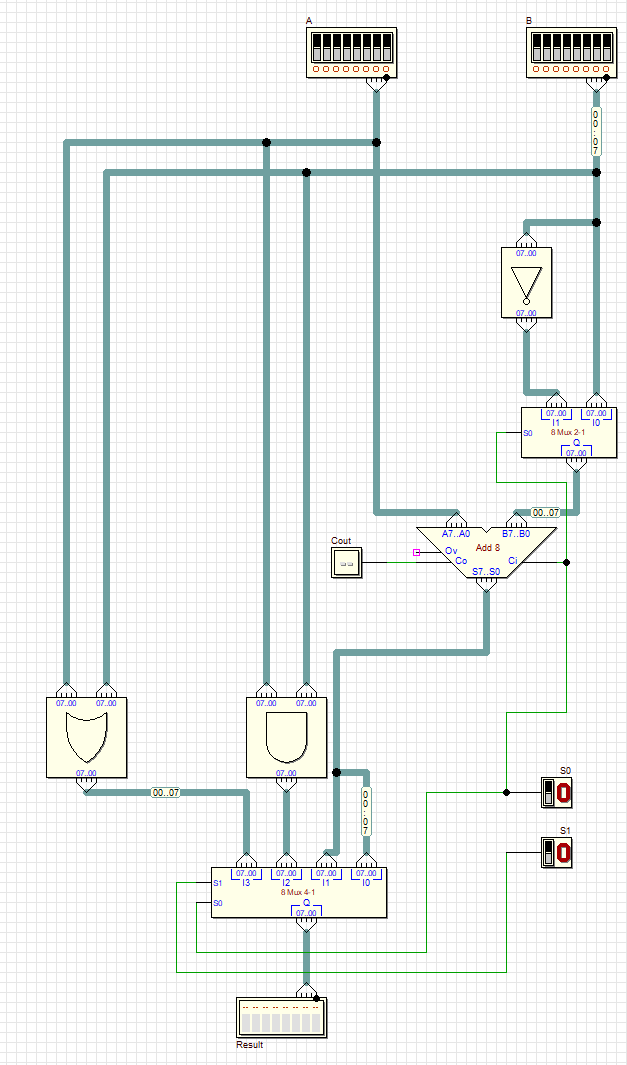
**Immagine che contiene tavolo

Descrizione generata automaticamente**

First, a simple version with 1-bit input A and B and 2-bit control signal. We import the components we have developed before (mux\_2\_bits.cbe, FullAdder.cbe and mux\_4\_bits.cbe) into the circuit to create the one-bit ALU.



Now, we can design a more complex version with 8-bits inputs and 2-bits control signal



Finally, as an exercise, you can try to design a different adder implementation (with a Carry-Lookahead Adder) and substitute the one on the 8-bit ALU.

