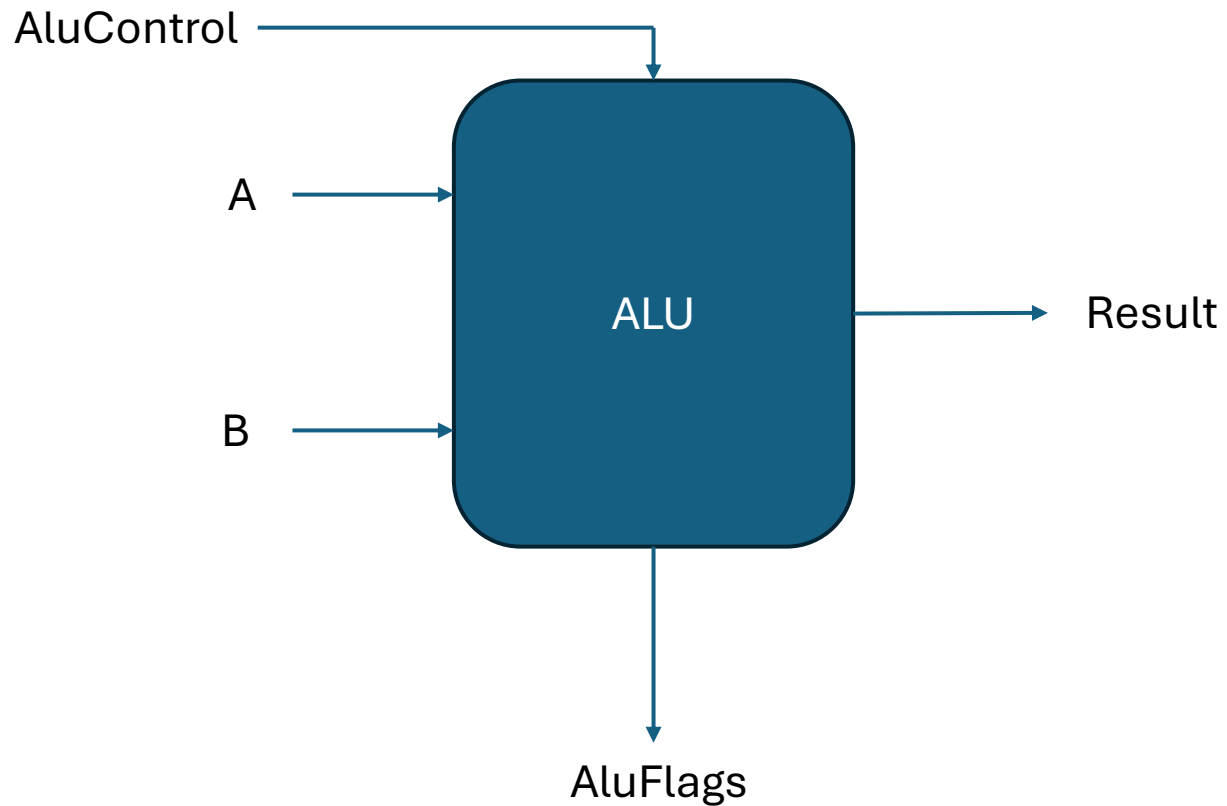


# Implementación de ALU en Basys3



### ***Pregunta 1: Simulación del ALU***

1. Con ayuda de las diapositivas implementar el diseño del ALU
2. Hacer un testbench para probar las cuatro operaciones:

1. 3+5
2. 5-5
3. 8 and 1
4. 5 or 7

Corroborar los resultados en el waveform y explicar las banderas obtenidas.

```

1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  # Clock signal
7  set_property PACKAGE_PIN W5 [get_ports clk]
8      set_property IOSTANDARD LVCMOS33 [get_ports clk]
9      create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
10
11  ## Switches
12  #set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
13  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
14  #set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
15  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
16  #set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
17  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
18  #set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
19  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
20  #set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
21  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
22  #set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
23  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
24  #set_property PACKAGE_PIN W14 [get_ports {sw[6]}]

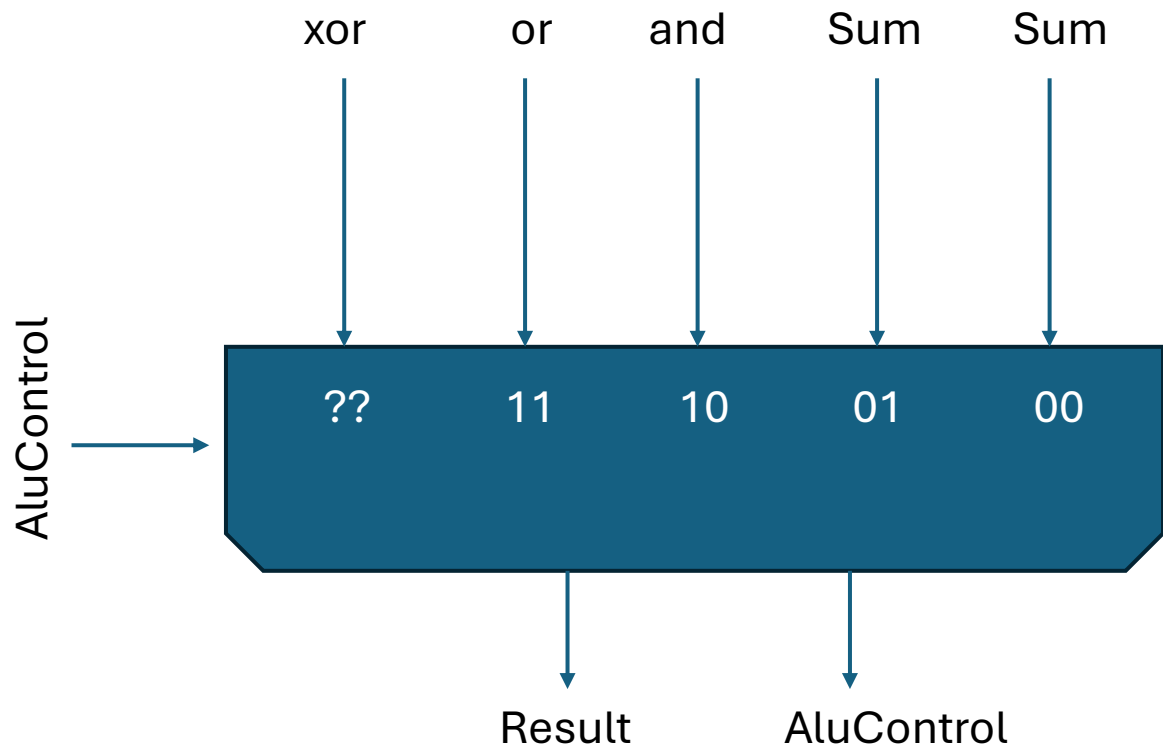
```

## Pregunta 2: Implementar el ALU en placa

1. Crear el archivo constraint usando el código que pueden encontrar en el repositorio de github:

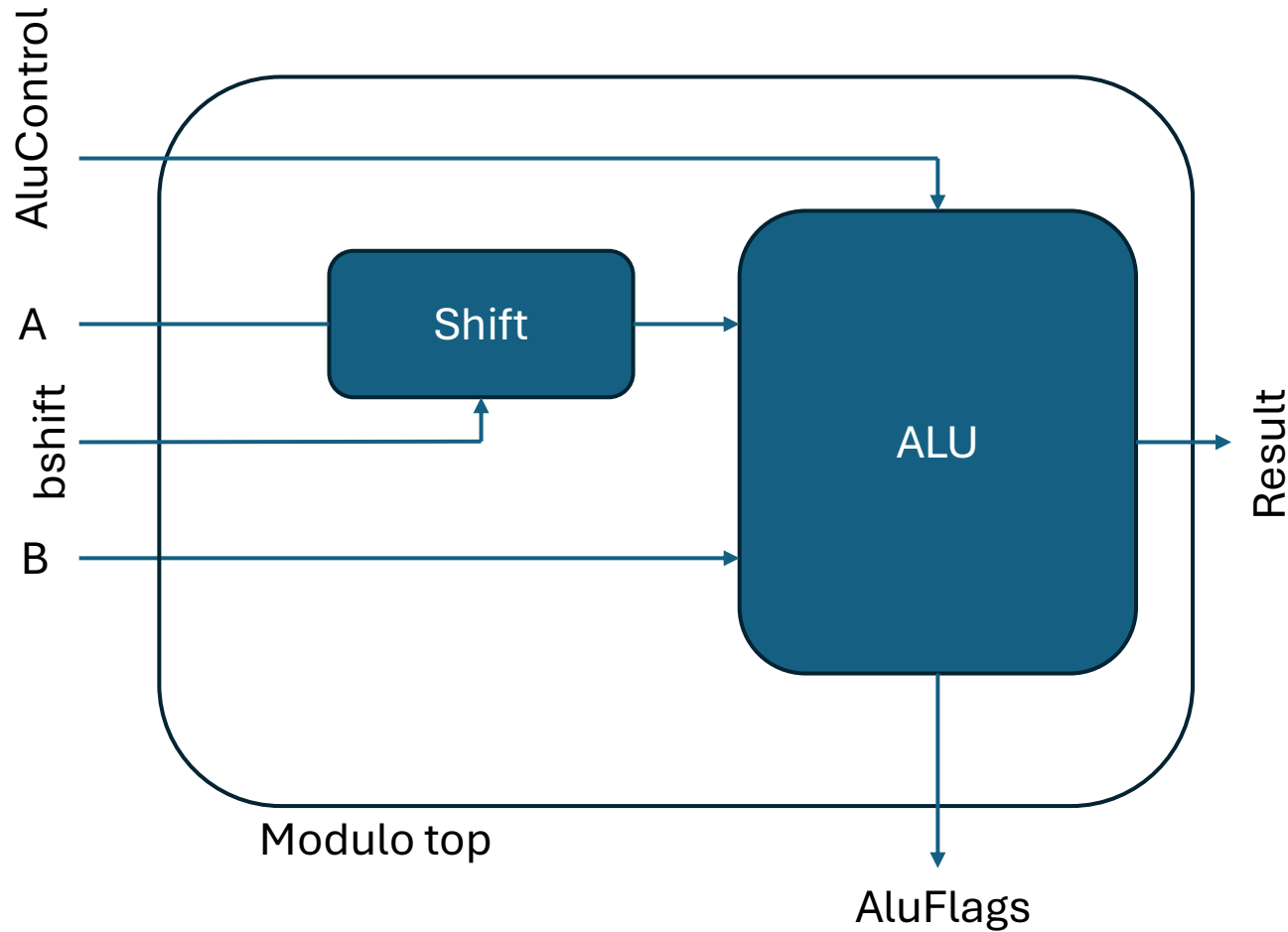
[https://github.com/Digilent/Basys3/blob/master/Projects/Keyboard/src/constraints/Basys3\\_Master.xdc](https://github.com/Digilent/Basys3/blob/master/Projects/Keyboard/src/constraints/Basys3_Master.xdc)

2. Como queremos utilizar los switches de la basys3 vamos a reducir el número de bits de nuestras entradas A y B a 5. Adaptar el módulo ALU.
3. Asignar los primeros 5 switches para el input A, los siguientes 5 switches para el input B y los últimos para el AluControl.
4. Asignar los primeros 5 leds para mostrar el result y los 4 últimos para mostrar los flags.
5. Generar el bitstream.
6. Conectar la placa y programarla.
7. Corroborar en la placa los casos de la pregunta 1.



**Pregunta 3: Modificar el ALU para que admita una nueva operación xor**

1. Implementar un ALU que admita la operación xor, para ello deberán modificar el modulo y la cantidad de bits del ALUControl. Trabajen sobre el ALU de 5 bits que previamente ya habían implementado.
2. Hacer un testbench para probar las 5 operaciones:
  1. 3+5
  2. 5-5
  3. 8 and 1
  4. 5 or 7
  5. 9 xor 6Corroborar los resultados en el waveform y explicar las banderas obtenidas.



**Pregunta 4: Agregar un modulo de shift a su diseño que permita realizar un shift de bits a la izquierda**

1. Implementar un modulo top que contenga los módulos shift y ALU. Trabajen sobre el ALU de 5 bits que previamente ya habían implementado con el xor.
2. Para implementar el módulo shift considere que tiene dos entradas A y bshift de 2 bits (que representa el número de bits que se va a desplazar la entrada A:  
[https://nandland.com/shift-operator/#google\\_vignette](https://nandland.com/shift-operator/#google_vignette)
3. Hagan un testbench que pruebe que su modulo top funciona correctamente.

```

1  ## This file is a general .xdc for the Basys3 rev B board
2  ## To use it in a project:
3  ## - uncomment the lines corresponding to used pins
4  ## - rename the used ports (in each line, after get_ports) according to the top level signal names in the project
5
6  # Clock signal
7  set_property PACKAGE_PIN W5 [get_ports clk]
8      set_property IOSTANDARD LVCMOS33 [get_ports clk]
9      create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]
10
11  ## Switches
12  #set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
13  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
14  #set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
15  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
16  #set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
17  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
18  #set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
19  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
20  #set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
21  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
22  #set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
23  #      set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
24  #set_property PACKAGE_PIN W14 [get_ports {sw[6]}]

```

## ***Pregunta 5: Implementar el modulo top en placa***

1. Crear el archivo constraint usando el codigo que pueden encontrar en el repositorio de github:

[https://github.com/Digilent/Basys3/blob/master/Projects/Keyboard/src/constraints/Basys3\\_Master.xdc](https://github.com/Digilent/Basys3/blob/master/Projects/Keyboard/src/constraints/Basys3_Master.xdc)

2. Asignar los primeros 5 switches para el input A, los siguientes 5 swithces para el input B y los ultimos para el AluControl y 2 para el bshift.
3. Asignar los primeros 5 leds para mostrar el result y los 4 ultimos para mostrar los flags.
4. Generar el bitstream.
5. Conectar la placa y programarla.
6. Corroborar en la placa los casos de la pregunta 4.