16 bit ALU

A screenshot of a computer

Description automatically generated

The ALU\_16bit module represents a 16-bit Arithmetic Logic Unit (ALU). It performs various arithmetic and logical operations based on the input signals A, B, and ALU\_Sel. The module has output signals ALU\_Ope, ALU\_Out, and CarryOut to provide the results of the ALU operations.

The ALU\_Ope output signal is a 16-bit bus that represents the result of the ALU operation. The specific value depends on the ALU\_Out signal. If ALU\_Out is true (1), ALU\_Ope will be set to 16'hFFFF (all ones); otherwise, it will be set to 16'h0000 (all zeros).

The CarryOut output signal represents the carry-out bit resulting from the ALU operation.

Inside the always block, the ALU\_Sel signal is used in a case statement to select the appropriate operation based on its value.

Here is a brief summary of the operations performed by the ALU based on ALU\_Sel:

4'b0000: Addition operation. The tmp variable holds the result of adding A and B. The carry-out bit is stored in CarryOut.

4'b0001: Bitwise OR operation. The tmp variable holds the result of performing a bitwise OR operation between A and B. The carry-out bit is stored in CarryOut.

4'b0010: Bitwise AND operation. The tmp variable holds the result of performing a bitwise AND operation between A and B. The carry-out bit is stored in CarryOut.

4'b0011: Left shift operation. The tmp variable holds the result of left-shifting A by 1 bit. The carry-out bit is stored in CarryOut.

4'b0100: Right shift operation. The tmp variable holds the result of right-shifting A by 1 bit. The carry-out bit is stored in CarryOut.

4'b0101: Subtraction operation. The tmp variable holds the result of subtracting B from A. The carry-out bit is stored in CarryOut. The ALU\_Out signal is set to indicate whether A is less than B.

4'b10xx: Multiplication operation. The tmp variable holds the result of multiplying A and B. The carry-out bit is stored in CarryOut.

4'b11xx: Division operation. The tmp variable holds the result of dividing A by B. The carry-out bit is stored in CarryOut.

Default: Bitwise copy operation. The tmp variable is assigned the value of B. The carry-out bit is stored in CarryOut.

The ALU\_Out and CarryOut signals are assigned inside the always block based on the selected operation.

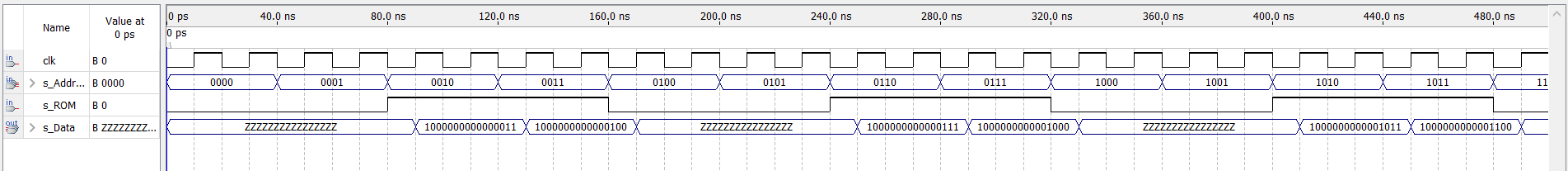
RAM

A screenshot of a computer

Description automatically generated

* On the positive edge of the clock (posedge clk), the RAM module performs its operations.
* If s\_RAM is active (high), indicating a write operation, the value from s\_Data is written into the memory at the specified s\_Address.
* If s\_read is active (high), indicating a read operation, the value at the specified s\_Address is retrieved from the memory and assigned to Data\_Output.
* If neither s\_RAM nor s\_read is active (both low), the Data\_Output is set to 'z' (high-impedance state), indicating an invalid or non-selected state.

ROM



* During the positive edge of the clock (posedge clk), if s\_ROM is active (high), the module reads the value corresponding to the s\_Address and assigns it to s\_Data.
* If s\_ROM is inactive (low), s\_Data is set to 'z' (high-impedance state).
* The default case in the case statement assigns s\_Data to 'z' if the s\_Address input does not match any specified cases.

Control Unit

A screenshot of a computer

Description automatically generated

* This always block is a combinational logic block that determines the control signals based on the current state. It updates the next\_state value based on the current state. Depending on the current state, it sets the appropriate control signals (c\_ROM, c\_RAM, and c\_read) accordingly.
* This always block is another combinational logic block that determines the value of the ALU selection (ALU\_Sel) based on the current state. It assigns the appropriate 4-bit value to ALU\_Sel depending on the current state.

Datapath Unit

A screenshot of a computer

Description automatically generated

* In the waveform, we will see the transitions of various signals, including the control signals (s\_Address, s\_ROM, s\_RAM, s\_read), register values (RegA to RegH), ALU operations (ALU\_Ope), data memory access (Data\_Output), and the traced address and data (Trace\_Addr and Trace\_Data).
* The waveform will show how these signals change and interact over time based on the clock and reset signals. The clock signal will drive the sequential execution of the datapath, and the reset signal will initialize the system to a known state.