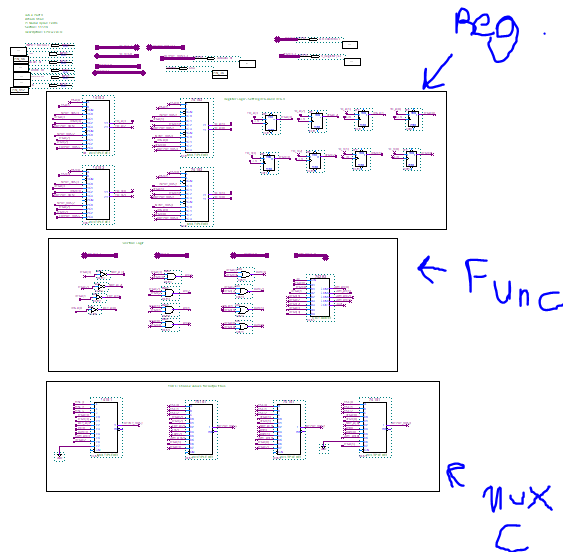
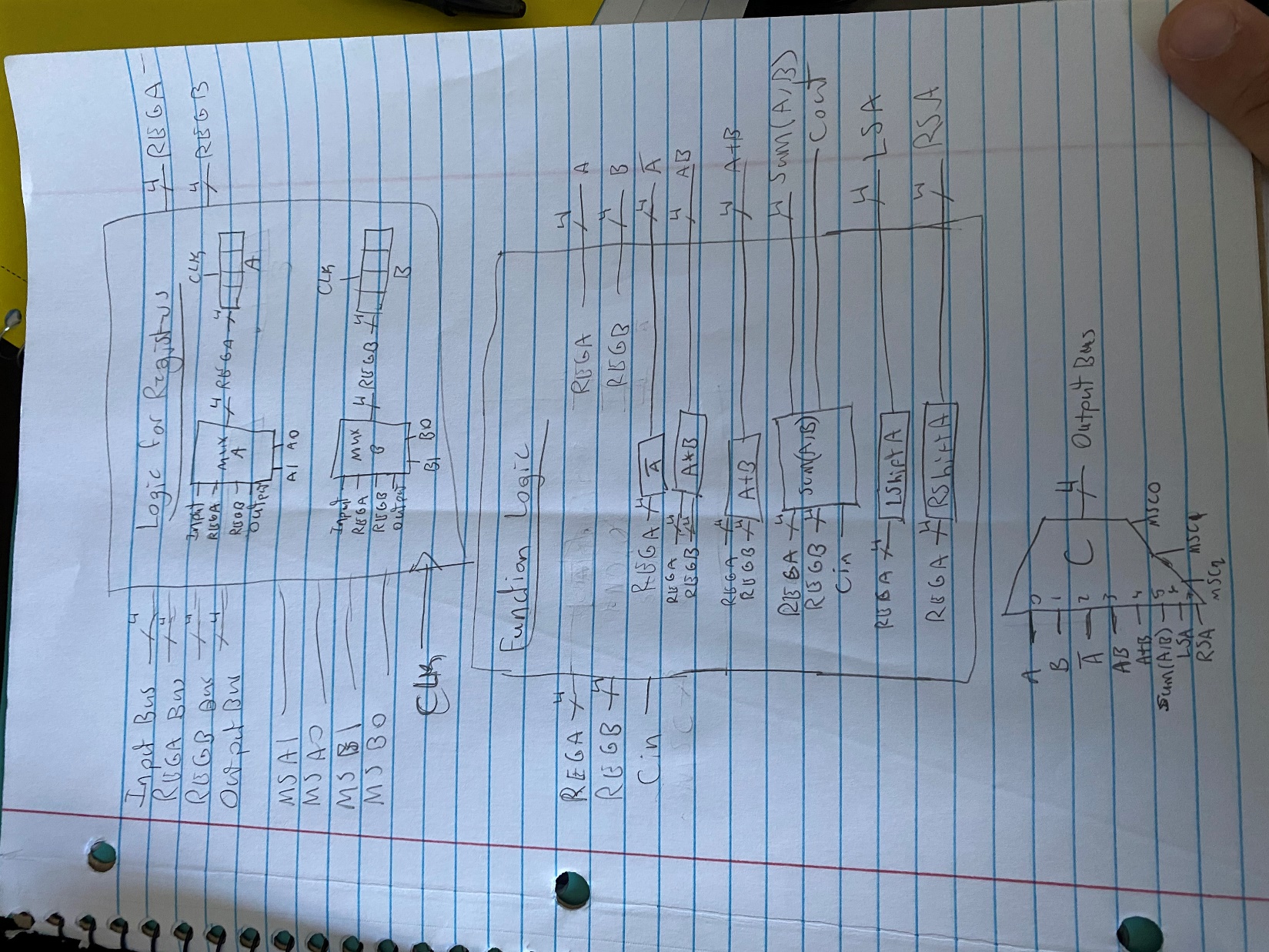
ALU and CPU design have obvious importance in computer engineering and various computing applications. But more so this lab serves to build foundational knowledge concerning how we might design systems which can make use of components such as an ALU. In particular the second portion of the lab makes it clear how one would define instructions in order to use registers for computations. The Output Bus contains the output from whatever operation is performed by the ALU.

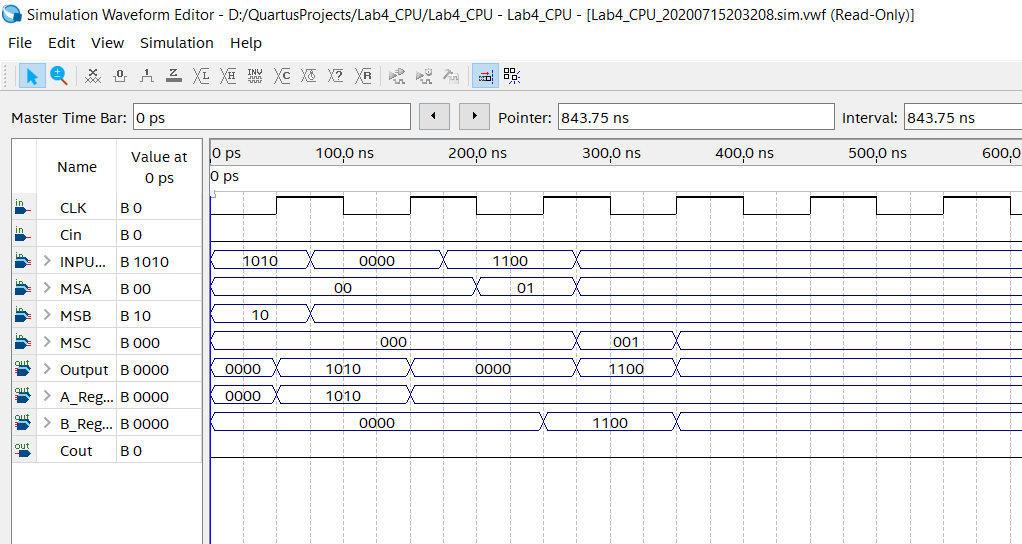
MSA, MSB, and MSC control the registers of the ALU as follows:

|  |  |  |
| --- | --- | --- |
| **MSA1/MSB1** | **MSA0/MSB0** | **Input to REGA/REGB** |
| 0 | 0 | Input Bus |
| 0 | 1 | REGA |
| 1 | 0 | REGB |
| 1 | 1 | Output Bus |

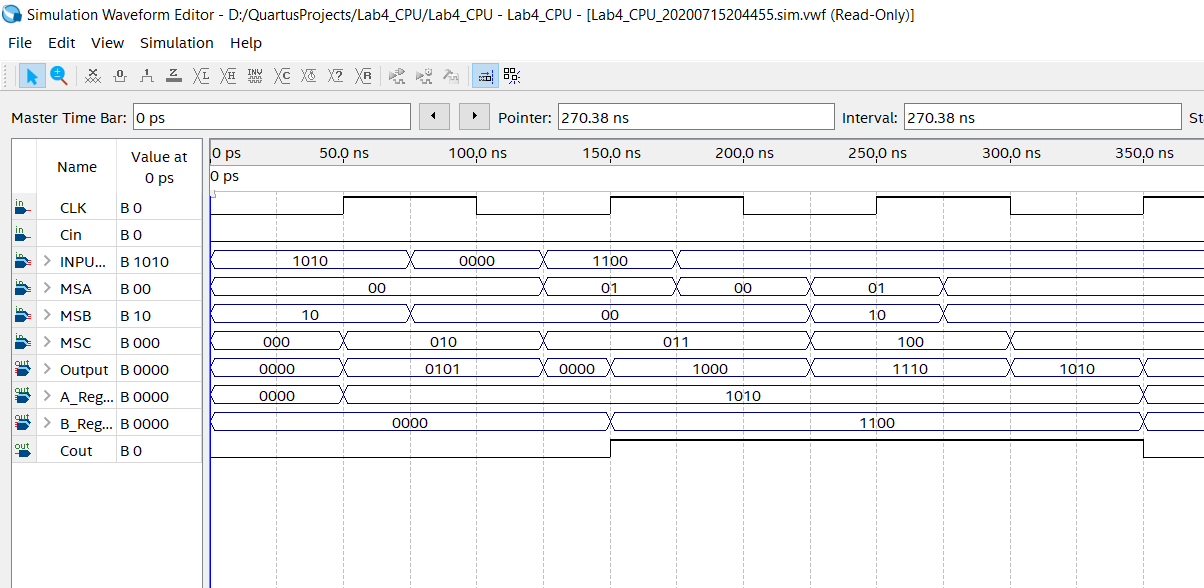
|  |  |  |  |
| --- | --- | --- | --- |
| **MSC2** | **MSC1** | **MSC0** | **Output Bus** |
| 0 | 0 | 0 | REGA |
| 0 | 0 | 1 | REGB |
| 0 | 1 | 0 | Complement REGA |
| 0 | 1 | 1 | Bitwise AND(A,B) |
| 1 | 0 | 0 | Bitwise OR(A,B) |
| 1 | 0 | 1 | Sum(A,B) |
| 1 | 1 | 0 | Left Shift A |
| 1 | 1 | 1 | Right Shift A |



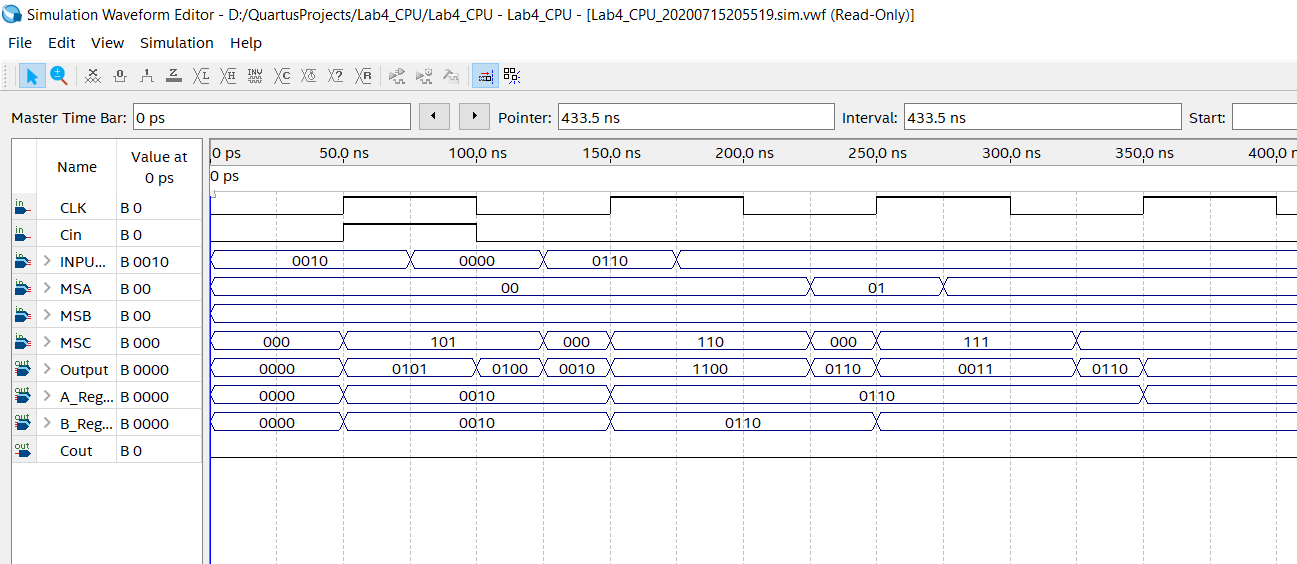
**Simulating Simple Functions:**



In the above, we test setting the output bus to REGA and REGB. First REGA is set to 1010 and this is copied to the output bus, then REGB is set to 1100 and this is copied to the output bus.



In the above, REGA is set to 1010 and then MSC is set to 010 to negate A, output is 0101. Then REGA maintains the same value and REGB is set to 1100, MSC is 011 (bit-AND) and the output is 1000. Then both REGA and REGB are the same and MSC is 100 (bit-OR) and the output is 1110.

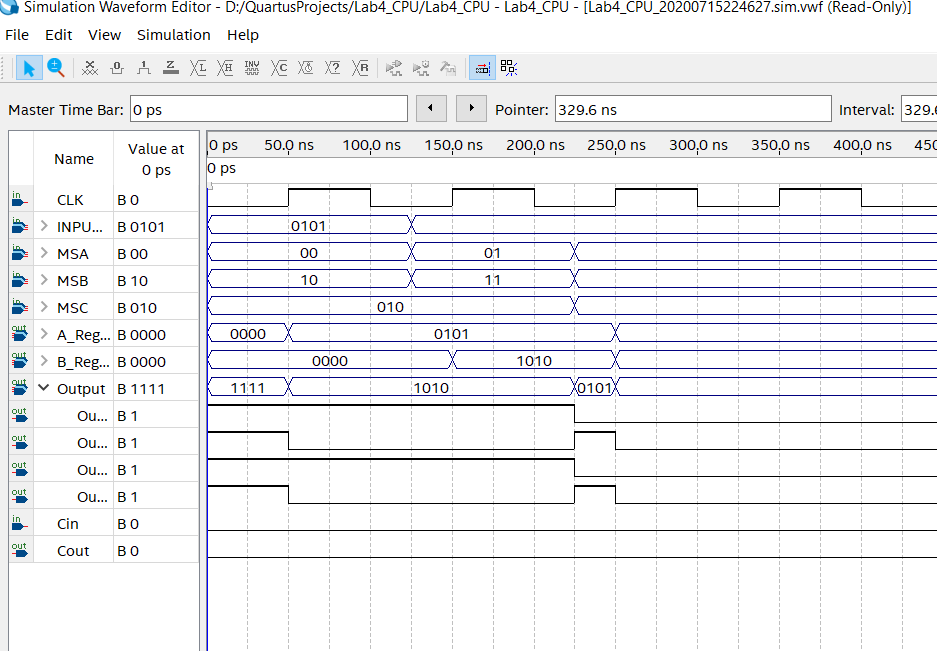


In the above, REGA and REGB are both set to 0010 and MSC is 101 (sum), when Cin=1 output =0101, when Cin=0 output=0100. Then REGA is set to 0110 and MSC is set to 110 (left shift), output is 1100. REGA maintains its value and MSC is set to 111 (right shift), output is 0011.

**4a:**

Load register A with data, complement the A data and then store it into B. Preserve the data in A during this operation

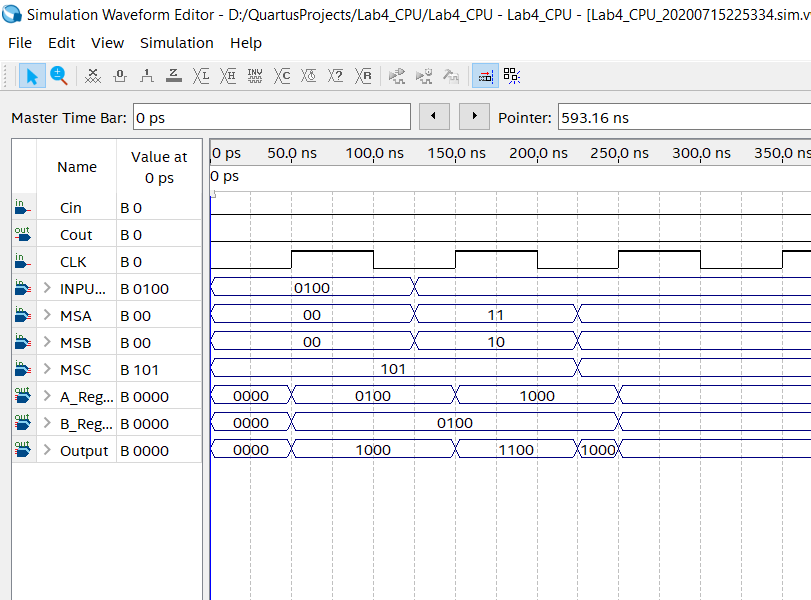
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MSA** | **MSB** | **MSC** | **Input** | **Cin** | **RegA** | **RegB** | **Output** | **RegA+** | **RegB+** | **Output+** | **Cout+** | **Description** |
| 00 | 10 | 010 | 0101 | 0 | 0000 | 0000 | 1111 | 0101 | 0000 | 1010 | 0 | Find /A |
| 01 | 11 | 010 | 0000 | 0 | 0101 | 0000 | 1010 | 0101 | 1010 | 1010 | 0 | B = /A |



**4b:**

Load A and B with known values, sum them and then store the result in A. Preserve B during this operation.

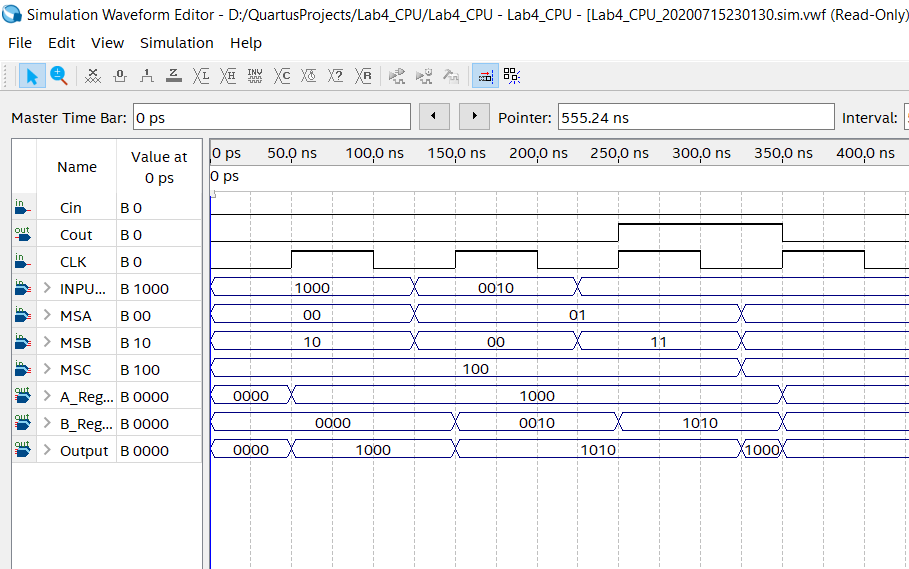
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MSA** | **MSB** | **MSC** | **Input** | **Cin** | **RegA** | **RegB** | **Output** | **RegA+** | **RegB+** | **Output+** | **Cout+** | **Description** |
| 00 | 00 | 101 | 0100 | 0 | 0000 | 0000 | 0000 | 0100 | 0100 | 1000 | 0 | Sum |
| 11 | 10 | 101 | 0000 | 0 | 0100 | 0100 | 1000 | 1000 | 0100 | 1100 | 0 | A = Sum |



**4c:**

Load A and B with known values, bit wise OR the registers and then store the result in B. Preserve the contents of register A during this operation.

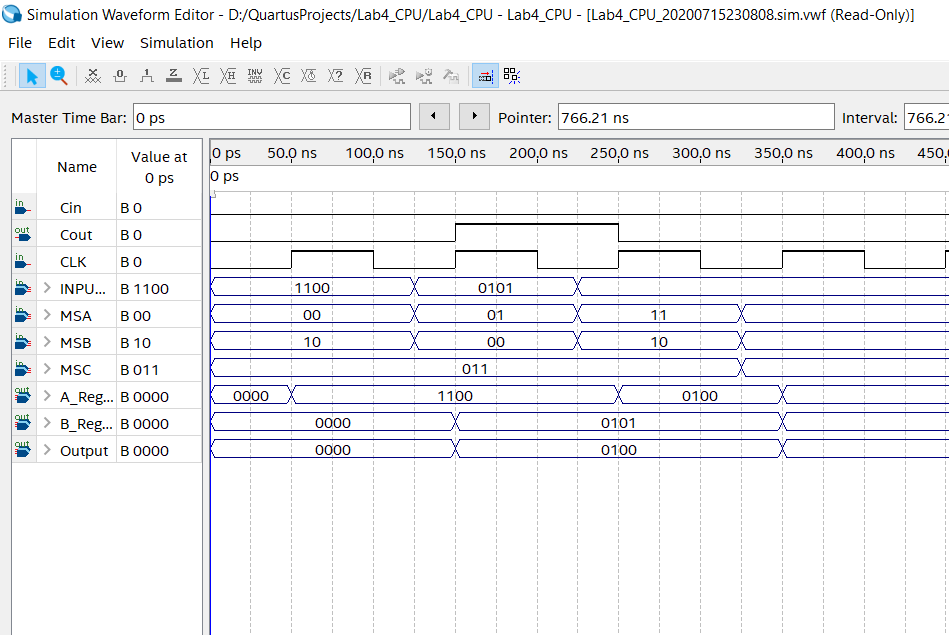
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MSA** | **MSB** | **MSC** | **Input** | **Cin** | **RegA** | **RegB** | **Output** | **RegA+** | **RegB+** | **Output+** | **Cout+** | **Description** |
| 00 | 10 | 100 | 1000 | 0 | 0000 | 0000 | 0000 | 1000 | 0000 | 1000 | 0 | Set A |
| 01 | 00 | 100 | 0010 | 0 | 1000 | 0000 | 1000 | 1000 | 0010 | 1010 | 0 | Set B and OR |
| 01 | 11 | 100 | 0000 | 0 | 1000 | 0010 | 1010 | 1000 | 1010 | 1010 | 0 | Store in B |



**4d:**

Load A and B with known values, bit wise AND the registers and then store the result in A. Preserve the contents of register B during this operation.

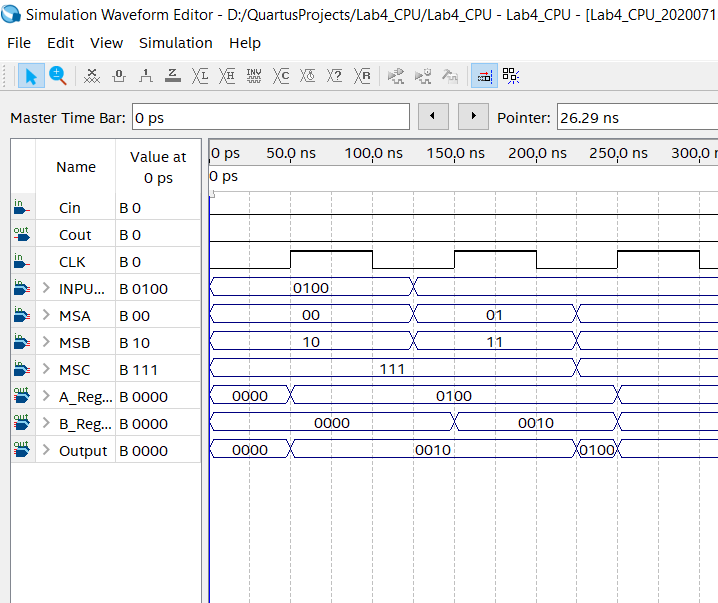
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MSA** | **MSB** | **MSC** | **Input** | **Cin** | **RegA** | **RegB** | **Output** | **RegA+** | **RegB+** | **Output+** | **Cout+** | **Description** |
| 00 | 10 | 011 | 1100 | 0 | 0000 | 0000 | 0000 | 1100 | 0000 | 0000 | 0 | Set A |
| 01 | 00 | 011 | 0101 | 0 | 1100 | 0000 | 0000 | 1100 | 0101 | 0100 | 0 | Set B and AND |
| 11 | 10 | 011 | 0000 | 0 | 1100 | 0101 | 0100 | 0100 | 0101 | 0100 | 0 | Store in A |



**4e:**

Load A with a value, shift it right one bit and then store it in B.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MSA** | **MSB** | **MSC** | **Input** | **Cin** | **RegA** | **RegB** | **Output** | **RegA+** | **RegB+** | **Output+** | **Cout+** | **Description** |
| 00 | 10 | 111 | 0100 | 0 | 0000 | 0000 | 0000 | 0100 | 0000 | 0010 | 0 | Set A and shift right |
| 01 | 11 | 111 | 0000 | 0 | 0100 | 0000 | 0010 | 0100 | 0010 | 0010 | 0 | Store in B |



**4f:**

OR 3 and 7, ADD 2 to the result, divide the result by 4, complement this new result, AND the result with 3, and then finally multiply it by 2. (Don’t worry if the result makes no sense; just perform the required operations. This “program” should work, independent of the inputs, i.e., the “program” should remain unchanged even if the 3 and 7 inputs are changed to $B and 6.)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MSA** | **MSB** | **MSC** | **Input** | **Cin** | **RegA** | **RegB** | **Output** | **RegA+** | **RegB+** | **Output+** | **Cout+** | **Description** |
| 00 | 10 | 100 | 0011 | 0 | 0000 | 0000 | 0000 | 0011 | 0000 | 0011 | 0 | A=3 |
| 01 | 00 | 100 | 0111 | 0 | 0011 | 0000 | 0011 | 0011 | 0111 | 0111 | 0 | B=7 O=A+B |
| 11 | 00 | 100 | 0010 | 0 | 0011 | 0111 | 0111 | 0111 | 0010 | 0111 | 0 | A = O, B = 2 |
| 01 | 10 | 101 | 0000 | 0 | 0111 | 0010 | 1001 | 0111 | 0010 | 1001 | 0 | O=Sum(A,B) |
| 11 | 00 | 101 | 0011 | 0 | 0111 | 0010 | 1001 | 1001 | 0011 | 1100 | 0 | A=O, B=3 |
| 11 | 10 | 111 | 0000 | 0 | 1001 | 0011 | 0100 | 0100 | 0011 | 0010 | 0 | A+=RS(A) |
| 11 | 10 | 111 | 0000 | 0 | 0100 | 0011 | 0010 | 0010 | 0011 | 0001 | 0 | A+=RS(A) |
| 11 | 10 | 010 | 0000 | 0 | 0010 | 0011 | 1101 | 1101 | 0011 | 0010 | 0 | A+=/A |
| 11 | 10 | 011 | 0000 | 0 | 1101 | 0011 | 0001 | 0001 | 0011 | 0001 | 0 | A+=A\*B |
| 11 | 10 | 110 | 0000 | 0 | 0001 | 0011 | 0010 | 0010 | 0011 | 0100 | 0 | A+=LS(A) |

