CompArch Lab 1

Jacob Riedel, Jennifer Vaccaro, Jennifer Wei

October 15 2015

1 Note to the PM

For this lab, we implemented a subset of the standard MIPS ALU. The number of operations supported has been reduced, but otherwise we are emulating that standard. Please find in our report a description of the implementation, the test results, and an analysis of the timing that demonstrate that our ALU design is complete, correct, and ready to be included in the CPU.

2 Implementation

We did not make any particularly inovative design choices. We tried to reuse logic, so the `add` module is nested within the `subtract` module, and the subtract module within our SLT. We incorporated bitslice logic whenever possible, and have included the schematics for the general bitwise logic when applicable.

Our complete ALU unit as shown in figure 1 uses a behavioral MUX to determine which function to output. Additionally, after selecting values for `result[31:0]`, `carryout`, and `overflow`, we use an nor gate to determine if these outputs are all false. If so, we set the output `zero` to true. This could be especially useful when using a subtraction function to determine if two inputs are equal.

2.1 ADD

For the 32-bit adder, we started with a single-bit adder module (figure 2. We then used the carryout from each full adder to yield the sum and carryout of the 32-bit adder (figure 3). We used the logic shown in figure 4, which checked whether the most significant result bit was yielding the correct sign. Because each single-bit adder takes its carryin from the carryout of the single-bit adder before it, this unit has a large time propagation delay.

2.2 SUB

The subtracter as shown in figure 5 was built off the 32-bit adder and an inverter for the second input B. We use the adder as a complete unit (no bitslice) but the inverter uses bitslice logic as shown in figure 6.

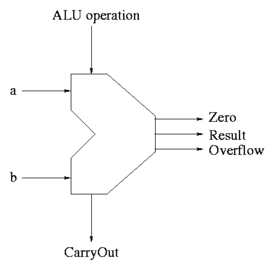


Figure 1: The ALU diagram from the lab README.

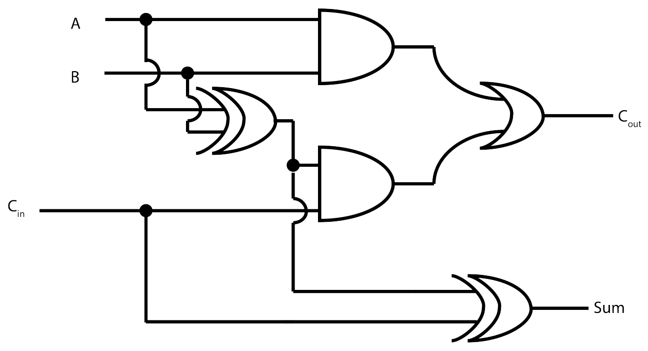


Figure 2: Single-bit adder

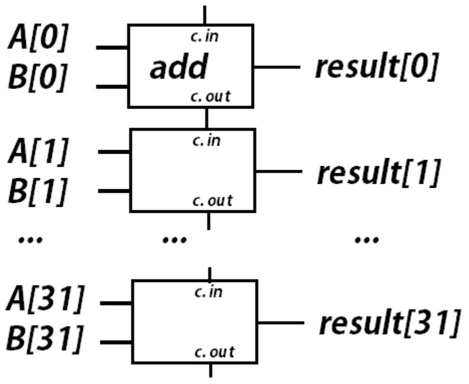


Figure 3: 32-bit adder outputs `result` and `carryout` depend on 32 single-bit adders in series.

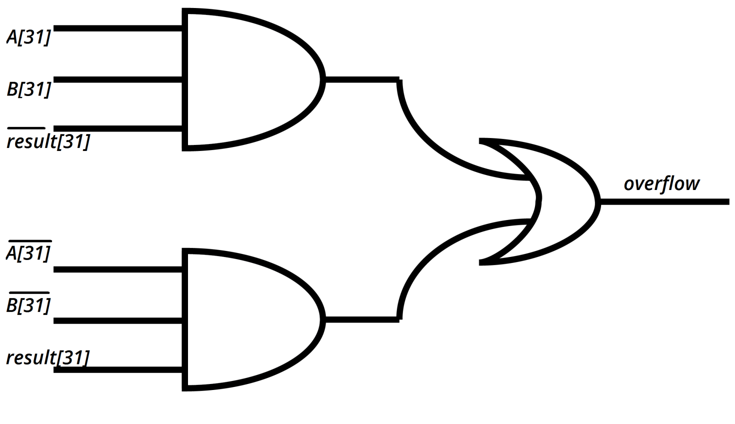


Figure 4: 32-bit adder output `overflow` checks whether `result[31]` displays the correct sign.

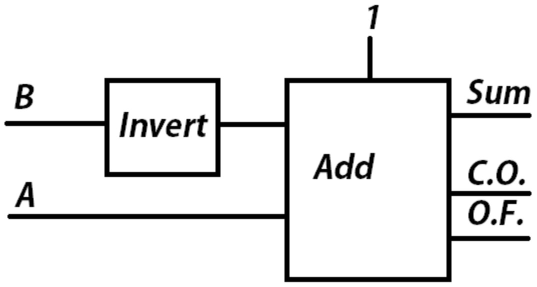


Figure 5: 32-bit SUBTRACT module incorporates the 32-bit adder and a bitwise inverter

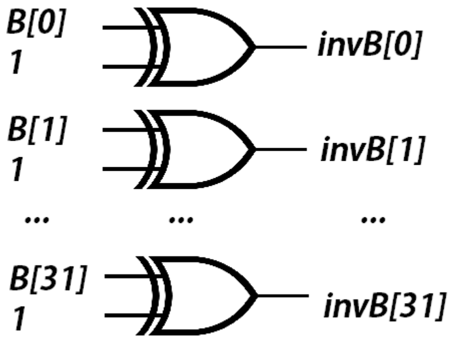


Figure 6: Bitwise inverter schematic

Additionally, to complete the transition to two's complement, we added a `carryin` = 1 input to the 32-bit adder unit.

2.3 XOR

Our XOR module used bitslice logic to determine the result, index by index, as shown in figure 7. Additionally, we set the other outputs: `carryout`, `overflow`, and `zero`: to zero.

2.4 SLT

In order to determine whether one input was larger than the other, we subtracted input `B` from input `A` and looked at the sign of the difference. We considered instead using a bit-by-bit comparison, since it would have been faster and potentially smaller, but implementation was simpler by nesting different module. In order to determine the positive/negative parity of the difference of the inputs, we either considered the highest magnitude bit `Dif[31]` xor with `overflow`. We determined that this was the logical equivalent of

$$\texttt{result[0]} = \overline{\texttt{overflow}}\texttt{Dif[31]} + \overline{\texttt{Dif[31]}} * \texttt{overflow} * \texttt{carryout}$$

Our SLT module was our slowest module, because it nested a subtract module and included additional logic in series. Additionally, we set the other outputs: `result[31:1]`, `carryout`, `overflow`, and `zero`: to zero.

2.5 AND

Our AND module used bitslice logic to determine the result, index by index, as shown in figure 9. Additionally, we set the other outputs: `carryout`, `overflow`, and `zero`: to zero.

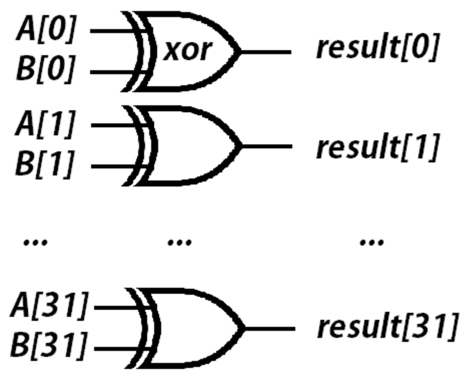


Figure 7: Bitwise XOR module schematic. Not included: all other outputs: `carryout`, `overflow`, `zero`: are zeroed.

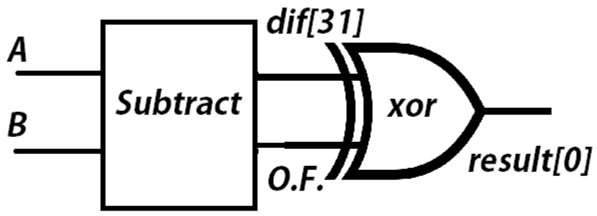


Figure 8: SLT module incorporates the 32-bit subtract module to compare two values. Not included: all other outputs: `carryout`, `overflow`, `zero`: are zeroed.

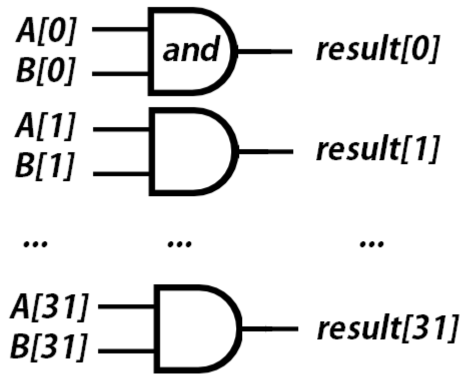


Figure 9: Bitwise AND module schematic. Not included: all other outputs: `carryout`, `overflow`, `zero`: are zeroed.

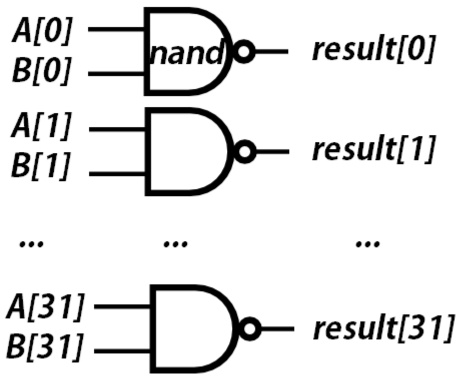


Figure 10: Bitwise NAND module schematic. Not included: all other outputs: `carryout`, `overflow`, `zero`: are zeroed.

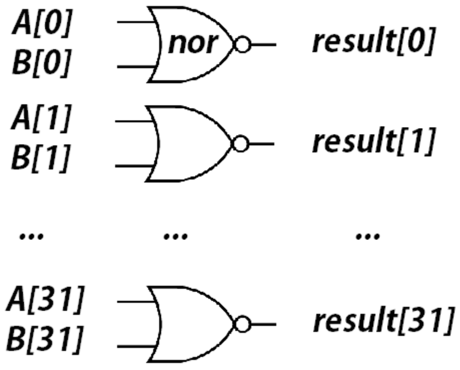


Figure 11: Bitwise NOR module schematic. Not included: all other outputs: `carryout`, `overflow`, `zero`: are zeroed.

2.6 NAND

Our NAND module used bitslice logic to determine the result, index by index, as shown in figure 10. Additionally, we set the other outputs: `carryout`, `overflow`, and `zero` to zero.

2.7 NOR

Our NOR module used bitslice logic to determine the result, index by index, as shown in figure 11. Additionally, we set the other outputs: `carryout`, `overflow`, and `zero` to zero.

2.8 OR

Our OR module used bitslice logic to determine the result, index by index, as shown in figure 12. Additionally, we set the other outputs: `carryout`, `overflow`, and `zero` to zero.

3 Test Results

Our test results are attached at the end of the document.

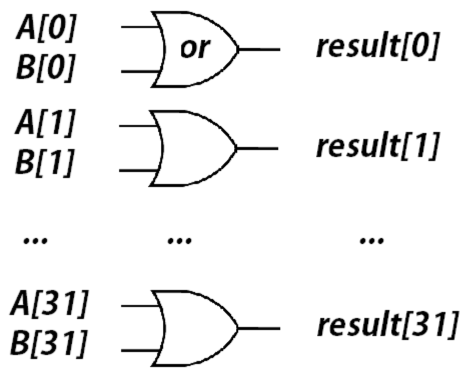


Figure 12: Bitwise OR module schematic. Not included: all other outputs: `carryout`, `overflow`, `zero`: are zeroed.

4 Test Benches

Test cases were chosen to include every type of output for every type of mathematical function. Because our ALU is designed for inputs given in 2's complement form, we chose a variety of high magnitude positive, high magnitude negative, low magnitude positive, low magnitude negative, and zero inputs. Additionally, we made certain that each flag was tested for each module.

Delays were the most difficult bugs in our design to fix. Because the four bit adder was a much smaller and faster unit, the delays we estimated were several orders of magnitude too small. Additionally, because the bitslice logic of the XOR, AND, NAND, NAND, and OR functions were much faster than the ADD, SUBTRACT, and SLT functions, we had to wait for all calculations to finish before selecting an output.

5 Timing Analysis

The longest running submodule will be the SLT, because it incorporates a bitslice, an adder, and two additional logic gates. Based on the given timing delays for this lab, the longest possible time propagation for the SLT will be 2630. However, because we built in a constant time delay of 10000 to our behavioral MUX in the ALU, and there is a $32 + 3$ port or gate after that, the worst case (and every case) delay for our ALU is 10010. If we wanted to run the alu faster, we could lower our constant time delay to 2630, for a total time delay of 2640 for the complete ALU unit.

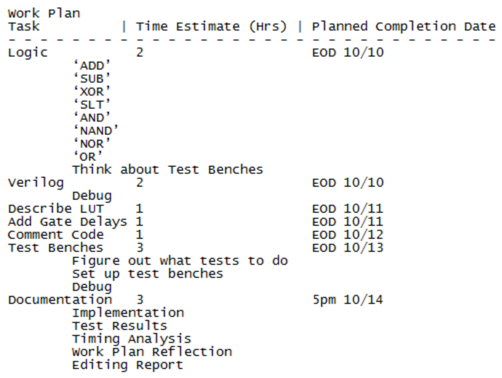
6 Work Plan Reflection

In our original work plan, we separated the tasks into seven bins: Logic, Verilog, Describe LUT, Add Gate Delays, Comment Code, Test Benches, and Documentation, where the sections that that took much longer than anticipated were the logic and the Verilog.

For the logic, we estimated that we would take two hours to figure out and draw out the logic for all the operands. We ended up taking about four hours overall, three initially and an additional hour to look over the more complex operands (i.e. `SLT` and `XOR`) since those required some modifications to the operands they built off of.

We estimated that the Verilog would take a couple hours since we thought that we would be able to just recycle that we had previously written, but integrating loops and more complex logic made it more time intensive, especially since we had not used loops before and had multiple team members coding in parallel, requiring additional time for integration and debugging.

Additionally, having members out of town at various stages of this lab made it more difficult. On the previous lab, we had multiple team meetings where we met in one location and worked together, so it was easy to stop and discuss any clarifications or confusions as they came up. This time, with people working



```
Work Plan
Task           | Time Estimate (Hrs) | Planned Completion Date
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Logic           2                     EOD 10/10
        'ADD'
        'SUB'
        'XOR'
        'SLT'
        'AND'
        'NAND'
        'NOR'
        'OR'
        Think about Test Benches
Verilog         2                     EOD 10/10
        Debug
Describe LUT    1                     EOD 10/11
Add Gate Delays 1                     EOD 10/11
Comment Code    1                     EOD 10/12
Test Benches    3                     EOD 10/13
        Figure out what tests to do
        Set up test benches
        Debug
Documentation   3                     5pm 10/14
        Implementation
        Test Results
        Timing Analysis
        Work Plan Reflection
        Editing Report
```

Figure 13: The work plan we wrote at the start of this lab.

somewhat separately, there were times when trying to debug or clarify something was difficult to do. However, we were flexible with our time and luckily had at least two of the three members at each meeting, which worked out.

7 Acknowledgments

Thank you to the NINJAs (Sophie, Sidd, Kyle, and Radmer) and Pratool for answering our questions about this lab and helping us better understand it.

```
#          OperandA                      OperandB            COM ||            Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  000 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  11111111111111111111111111111111  000 || 11111111111111111111111111111110 ||  1  0  0  | 11111111111111111111111111111110
# 00000000000000000000000000000001  00000000000000000000000000000001  000 || 00000000000000000000000000000010 ||  0  0  1  | 00000000000000000000000000000010
# 10011111111111111111111111111111  10011111111111111111111111111111  000 || 00111111111111111111111111111110 ||  1  1  0  | 00111111111111111111111111111110
# 01110000000000000000000000000000  01110000000000000000000000000000  000 || 11100000000000000000000000000000 ||  0  1  0  | 11100000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000000  000 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  00000000000000000000000000000001  000 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  11111111111111111111111111111111  000 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000000  000 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000001  000 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 00000000000000000000000000000001  11111111111111111111111111111111  000 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 11111111111111111111111111111001  00000000000000000000000000000001  000 || 11111111111111111111111111111010 ||  0  0  1  | 11111111111111111111111111111010
# 00010000000000000000000000000000  11111111111111111111111111111001  000 || 00001111111111111111111111111001 ||  1  0  0  | 00001111111111111111111111111001
# 01110000000000000000000000000000  11111111111111111111111111111111  000 || 01101111111111111111111111111111 ||  1  0  0  | 01101111111111111111111111111111
# 11111111111111111111111111111111  01110000000000000000000000000000  000 || 01101111111111111111111111111111 ||  1  0  0  | 01101111111111111111111111111111
# 01110000000000000000000000000000  10011111111111111111111111111111  000 || 00001111111111111111111111111111 ||  1  0  0  | 00001111111111111111111111111111
# 10011111111111111111111111111111  01110000000000000000000000000000  000 || 00001111111111111111111111111111 ||  1  0  0  | 00001111111111111111111111111111
#          OperandA                      OperandB            COM ||            Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  001 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 11111111111111111111111111111111  11111111111111111111111111111111  001 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000001  001 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 10011111111111111111111111111111  10011111111111111111111111111111  001 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 01110000000000000000000000000000  01110000000000000000000000000000  001 || 00000000000000000000000000000000 ||  1  0  0  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000000  001 || 00000000000000000000000000000001 ||  1  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  00000000000000000000000000000001  001 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 00000000000000000000000000000000  11111111111111111111111111111111  001 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 11111111111111111111111111111111  00000000000000000000000000000000  001 || 11111111111111111111111111111111 ||  1  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000001  001 || 11111111111111111111111111111110 ||  1  0  0  | 11111111111111111111111111111110
# 00000000000000000000000000000001  11111111111111111111111111111111  001 || 00000000000000000000000000000010 ||  0  0  1  | 00000000000000000000000000000010
# 11111111111111111111111111111001  00000000000000000000000000000001  001 || 11111111111111111111111111111000 ||  1  0  0  | 11111111111111111111111111111000
# 00010000000000000000000000000000  11111111111111111111111111111001  001 || 00010000000000000000000000000111 ||  0  0  0  | 00010000000000000000000000000111
# 01110000000000000000000000000000  11111111111111111111111111111111  001 || 01110000000000000000000000000001 ||  0  0  0  | 01110000000000000000000000000001
# 11111111111111111111111111111111  01110000000000000000000000000000  001 || 10001111111111111111111111111111 ||  1  0  0  | 10001111111111111111111111111111
# 01110000000000000000000000000000  10011111111111111111111111111111  001 || 11010000000000000000000000000001 ||  0  1  0  | 11010000000000000000000000000001
# 10011111111111111111111111111111  01110000000000000000000000000000  001 || 00101111111111111111111111111111 ||  1  1  0  | 00101111111111111111111111111111
#          OperandA                      OperandB            COM ||            Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  010 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  11111111111111111111111111111111  010 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000001  010 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 10011111111111111111111111111111  10011111111111111111111111111111  010 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 01110000000000000000000000000000  01110000000000000000000000000000  010 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000000  010 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  00000000000000000000000000000001  010 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  11111111111111111111111111111111  010 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000000  010 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000001  010 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 00000000000000000000000000000001  11111111111111111111111111111111  010 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 11111111111111111111111111111001  00000000000000000000000000000001  010 || 11111111111111111111111111111000 ||  0  0  1  | 11111111111111111111111111111000
# 00010000000000000000000000000000  11111111111111111111111111111001  010 || 11101111111111111111111111111001 ||  0  0  0  | 11101111111111111111111111111001
# 01110000000000000000000000000000  11111111111111111111111111111111  010 || 10001111111111111111111111111111 ||  0  0  0  | 10001111111111111111111111111111
# 11111111111111111111111111111111  01110000000000000000000000000000  010 || 10001111111111111111111111111111 ||  0  0  0  | 10001111111111111111111111111111
# 01110000000000000000000000000000  10011111111111111111111111111111  010 || 11101111111111111111111111111111 ||  0  0  0  | 11101111111111111111111111111111
# 10011111111111111111111111111111  01110000000000000000000000000000  010 || 11101111111111111111111111111111 ||  0  0  0  | 11101111111111111111111111111111
#          OperandA                      OperandB            COM ||            Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  11111111111111111111111111111111  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000001  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 10011111111111111111111111111111  10011111111111111111111111111111  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 01110000000000000000000000000000  01110000000000000000000000000000  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000000  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000000  00000000000000000000000000000001  011 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  11111111111111111111111111111111  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  00000000000000000000000000000000  011 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 11111111111111111111111111111111  00000000000000000000000000000001  011 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000001  11111111111111111111111111111111  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111001  00000000000000000000000000000001  011 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00010000000000000000000000000000  11111111111111111111111111111001  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 01110000000000000000000000000000  11111111111111111111111111111111  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  01110000000000000000000000000000  011 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 01110000000000000000000000000000  10011111111111111111111111111111  011 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 10011111111111111111111111111111  01110000000000000000000000000000  011 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
#          OperandA                      OperandB            COM ||            Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  100 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  11111111111111111111111111111111  100 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 00000000000000000000000000000001  00000000000000000000000000000001  100 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 10011111111111111111111111111111  10011111111111111111111111111111  100 || 10011111111111111111111111111111 ||  0  0  0  | 10011111111111111111111111111111
# 01110000000000000000000000000000  01110000000000000000000000000000  100 || 01110000000000000000000000000000 ||  0  0  1  | 01110000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000000  100 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000000  00000000000000000000000000000001  100 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000000  11111111111111111111111111111111  100 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  00000000000000000000000000000000  100 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  00000000000000000000000000000001  100 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000001  11111111111111111111111111111111  100 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 11111111111111111111111111111001  00000000000000000000000000000001  100 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00010000000000000000000000000000  11111111111111111111111111111001  100 || 00010000000000000000000000000000 ||  0  0  1  | 00010000000000000000000000000000
# 01110000000000000000000000000000  11111111111111111111111111111111  100 || 01110000000000000000000000000000 ||  0  0  1  | 01110000000000000000000000000000
# 11111111111111111111111111111111  01110000000000000000000000000000  100 || 01110000000000000000000000000000 ||  0  0  1  | 01110000000000000000000000000000
# 01110000000000000000000000000000  10011111111111111111111111111111  100 || 00010000000000000000000000000000 ||  0  0  1  | 00010000000000000000000000000000
# 10011111111111111111111111111111  01110000000000000000000000000000  100 || 00010000000000000000000000000000 ||  0  0  1  | 00010000000000000000000000000000
#           OperandA                          OperandB            COM ||             Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  101 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  11111111111111111111111111111111  101 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000001  101 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 10011111111111111111111111111111  10011111111111111111111111111111  101 || 01100000000000000000000000000000 ||  0  0  1  | 01100000000000000000000000000000
# 01110000000000000000000000000000  01110000000000000000000000000000  101 || 10001111111111111111111111111111 ||  0  0  0  | 10001111111111111111111111111111
# 00000000000000000000000000000001  00000000000000000000000000000000  101 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 00000000000000000000000000000000  00000000000000000000000000000001  101 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 00000000000000000000000000000000  11111111111111111111111111111111  101 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000000  101 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000001  101 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 00000000000000000000000000000001  11111111111111111111111111111111  101 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 11111111111111111111111111111001  00000000000000000000000000000001  101 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 00010000000000000000000000000000  11111111111111111111111111111001  101 || 11101111111111111111111111111111 ||  0  0  0  | 11101111111111111111111111111111
# 01110000000000000000000000000000  11111111111111111111111111111111  101 || 10001111111111111111111111111111 ||  0  0  0  | 10001111111111111111111111111111
# 11111111111111111111111111111111  01110000000000000000000000000000  101 || 10001111111111111111111111111111 ||  0  0  0  | 10001111111111111111111111111111
# 01110000000000000000000000000000  10011111111111111111111111111111  101 || 11101111111111111111111111111111 ||  0  0  0  | 11101111111111111111111111111111
# 10011111111111111111111111111111  01110000000000000000000000000000  101 || 11101111111111111111111111111111 ||  0  0  0  | 11101111111111111111111111111111
#           OperandA                          OperandB            COM ||             Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  110 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  11111111111111111111111111111111  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000001  110 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 10011111111111111111111111111111  10011111111111111111111111111111  110 || 01100000000000000000000000000000 ||  0  0  1  | 01100000000000000000000000000000
# 01110000000000000000000000000000  01110000000000000000000000000000  110 || 10001111111111111111111111111111 ||  0  0  0  | 10001111111111111111111111111111
# 00000000000000000000000000000001  00000000000000000000000000000000  110 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 00000000000000000000000000000000  00000000000000000000000000000001  110 || 11111111111111111111111111111110 ||  0  0  1  | 11111111111111111111111111111110
# 00000000000000000000000000000000  11111111111111111111111111111111  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  00000000000000000000000000000000  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  00000000000000000000000000000001  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 00000000000000000000000000000001  11111111111111111111111111111111  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111001  00000000000000000000000000000001  110 || 00000000000000000000000000000110 ||  0  0  1  | 00000000000000000000000000000110
# 00010000000000000000000000000000  11111111111111111111111111111001  110 || 00000000000000000000000000000110 ||  0  0  1  | 00000000000000000000000000000110
# 01110000000000000000000000000000  11111111111111111111111111111111  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  01110000000000000000000000000000  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 01110000000000000000000000000000  10011111111111111111111111111111  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 10011111111111111111111111111111  01110000000000000000000000000000  110 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
#           OperandA                          OperandB            COM ||             Result            ||  Co Of Zr | Expected Output
# 00000000000000000000000000000000  00000000000000000000000000000000  111 || 00000000000000000000000000000000 ||  0  0  1  | 00000000000000000000000000000000
# 11111111111111111111111111111111  11111111111111111111111111111111  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 00000000000000000000000000000001  00000000000000000000000000000001  111 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 10011111111111111111111111111111  10011111111111111111111111111111  111 || 10011111111111111111111111111111 ||  0  0  0  | 10011111111111111111111111111111
# 01110000000000000000000000000000  01110000000000000000000000000000  111 || 01110000000000000000000000000000 ||  0  0  1  | 01110000000000000000000000000000
# 00000000000000000000000000000001  00000000000000000000000000000000  111 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  00000000000000000000000000000001  111 || 00000000000000000000000000000001 ||  0  0  0  | 00000000000000000000000000000001
# 00000000000000000000000000000000  11111111111111111111111111111111  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000000  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  00000000000000000000000000000001  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 00000000000000000000000000000001  11111111111111111111111111111111  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111001  00000000000000000000000000000001  111 || 11111111111111111111111111111001 ||  0  0  0  | 11111111111111111111111111111001
# 00010000000000000000000000000000  11111111111111111111111111111001  111 || 11111111111111111111111111111001 ||  0  0  0  | 11111111111111111111111111111001
# 01110000000000000000000000000000  11111111111111111111111111111111  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 11111111111111111111111111111111  01110000000000000000000000000000  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 01110000000000000000000000000000  10011111111111111111111111111111  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
# 10011111111111111111111111111111  01110000000000000000000000000000  111 || 11111111111111111111111111111111 ||  0  0  0  | 11111111111111111111111111111111
```