Nema Karimi and David Landry  
B EE 271 A, Lab 4

**State Calculator**

For our final lab project, we designed a state calculator that is able to compute 4 functions: add, subtract, multiply, and divide. Inputs and results are 24 bits long (6 hex digits). For multiplication, we have an extended accumulator that holds an additional 24 bits. For division, we have a 24-bit long remainder as well.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Description** | **Input 1** | **Input 2** | **Expected Result** | **Actual Result** | **Notes** |
| **Addition** | | | | | |
| Addition by 0 | 7 | 0 | 7 | 7 | Passed |
| Normal addition | 9 | 7 | 10 | 10 | Passed |
| Overflow | FFFFFF | 1 | 0 | 0 | Passed |
| **Subtraction** | | | | | |
| Subtract by 0 | 10 | 0 | 10 | 10 | Passed |
| Subtract 0 by a number | 0 | 1 | FFFFFF | FFFFFF | Passed |
| Normal subtraction | 10 | 1 | F | F | Passed |
| **Multiplication** | | | | | |
| Multiply by 0: # x 0 | 5 | 0 | 0 | 0 | Passed |
| Multiply by 0: 0 x # | 0 | 5 | 0 | 0 | Passed |
| Normal multiplication | 5 | 2 | A | 0 | There is an issue with blocking vs. non-blocking statements. |
| Retest normal multiplication after code fix | 5 | 2 | A | A | Passed; fix worked. |
| Extended multiplication | 100000 | 100000 | 10000\_000000 | 10000\_000000 | Passed |
| **Division** | | | | | |
| Normal division, no remainder | 6 | 2 | 3, R0 | 3, R0 | Passed |
| Normal division, remainder | 9 | 2 | 4, R1 | 4, R1 | Passed |
| Division, small by large | 2 | 9 | 0, R2 | 0, R2 | Passed |
| Division by 0 | 5 | 0 | FFFFFF, R5 | 5, R5 | Accidentally changed code to bypass division by 0. |
| Division by 0 after fix | 5 | 0 | FFFFFF, R5 | FFFFFF, R5 | Passed; fix worked. |