Test Cases for ALU

ADDER:

Sum

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 0 | Positive + Positive  No overflow | 0x0001 | 0x0002 | 0x0003  Z = 0  V = 0  N = 0 |
| 1 | Positive + Positive  Overflow | 0x7FFF | 0x7FFF | 0xFFFE  Z = 0  V = 1  N = 1 |
| 2 | Positive + Negative | 0x0003 | 0xFFFF | 0x0002  Z = 0  V = 0  N = 0 |
| 3 | Negative + Negative  Overflow | 0xF000 | 0xF000 | 0xE000  Z = 0  V = 1  N = 1 |

Subtract

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 4 | Positive – Positive  No overflow | 0x0003 | 0x0002 | 0x0001  Z = 0  V = 0  N = 0 |
| 5 | Positive – negative  No overflow | 0x0001 | 0xFFFF | 0x0002  Z = 0  V = 0  N = 0 |
| 6 | Positive – negative  Overflow | 0x7FFF | 0xFFFF | 0x8000  Z = 0  V = 1  N = 1 |
| 7 | Negative – Negative | 0xF004 | 0xF005 | 0xFFFF  Z = 0  V = 0  N = 1 |

Multiply

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 8 | Positive \* Positive  No overflow | 0x0003 | 0x0002 | 0x0006  Z = 0  V = 0  N = 0 |
| 9 | Positive \* positive  Overflow | 0x7FFF | 0x7FFF | 0x0001  Z = 0  V = 1  N =0 |
| 10 | Positive \* Negative | 0x0001 | 0xFFFF | 0xFFFF  Z = 0  V = 0  N = 1 |
| 11 | 0 \* 0 | 0x0000 | 0x0000 | 0x0000  Z = 1  V = 0  N = 0 |
| 12 | Negative \* Negative | 0xFFFF | 0xFFFA | 0x0006  Z = 0  V = 1  N =0 |

Divide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 13 | Positive / Positive | 0x0004 | 0x0002 | 0x0002  Z = 0  V = 0  N = 0 |
| 14 | Positive / Negative | 0x0004 | 0xFFFE | 0xFFFE  Z = 0  V = 0  N = 1 |
| 15 | Negative / Negative | 0xFFFC | 0xFFFE | 0x0002  Z = 0  V = 0  N = 0 |

COMPARE:

More Than Equal (ALUFN 00)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 16 | A = B | 0x0001 | 0x0001 | 0x0001 |
| 17 | A > B | 0x0002 | 0x0001 | 0x0001 |
| 18 | A < B | 0x0001 | 0x0002 | 0x0000 |

Equal (ALUFN 01)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number |  |  |  |  |
| 19 | A = B | 0x0003 | 0x0003 | 0x0001 |
| 20 | A != B | 0x0001 | 0x0002 | 0x0000 |

Less than (ALUFN 10)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 21 | A = B | 0x0001 | 0x0001 | 0x0000 |
| 22 | A > B | 0x0002 | 0x0001 | 0x0000 |
| 23 | A < B | 0x0001 | 0x0002 | 0x0001 |

Less Than Equal (ALUFN 11)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 24 | A = B | 0x0001 | 0x0001 | 0x0001 |
| 25 | A > B | 0x0002 | 0x0001 | 0x0000 |
| 26 | A < B | 0x0001 | 0x0002 | 0x0001 |

BOOLEAN:

And

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 27 | A and B are opposite | 0101010101010101 | 1010101010101010 | 0000000000000000 |
| 28 | B is given a random value | 0101010101010101 | 1111000011110000 | 0101000001010000 |

Or

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 29 | A and B are opposite | 0101010101010101 | 1010101010101010 | 1111111111111111 |
| 30 | B is given a random value | 0101010101010101 | 1111000011110000 | 1111010111110101 |

XOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 31 | A and B are opposite | 0101010101010101 | 1010101010101010 | 1111111111111111 |
| 32 | B is given a random value | 0101010101010101 | 1111000011110000 | 1010010110100101 |

LDR(“A”)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 33 | A and B are opposite | 0101010101010101 | 1010101010101010 | 0101010101010101 |
| 34 | B is given a random value | 0101010101010101 | 1111000011110000 | 0101010101010101 |

SHIFTER:

Shift Left

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 35 | A is smallest negative number to shift by 15 bits | 1111111111111111 | 1111 | 1000000000000000 |
| 36 | A is second smallest negative number to shift by 15 bits | 1111111111111110 | 1111 | 0000000000000000 |
| 37 | Random A shift by 2 bits | 1010101010101010 | 0010 | 1010101010101000 |

Shift Right

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 38 | A is smallest negative number to shift by 15 bits | 1111111111111111 | 1111 | 0000000000000001 |
| 39 | A is largest positive number to shift by 15 bits | 0111111111111111 | 1111 | 0000000000000000 |
| 40 | Random A shift by 2 bits | 1010101010101010 | 0010 | 0010101010101010 |

Shift Right Arithmetic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | Test Case | A | B | Result |
| 41 | Random negative A to shift by 15 bits | 1010101010101010 | 1111 | 1111111111111111 |
| 42 | Random positive A to shift by 15 bits | 0101010101010101 | 1111 | 0000000000000000 |

ERROR:

We test for error by trying to shift left by 16 bits. However, as our ALU only takes in B[4:0], it will shift A by 0 bits instead.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case Number | A | B | Expected Result | ALU Result |
| 43 | 1111111111111111 | 10000 | 0000000000000000 | 1111111111111111 |