

Verification Plan

Jacob Halsey

Addition and Subtraction

ADD and SUB need to be tested across a range of inputs and for each port. The result data must be checked that it matches the expected result of the calculation, and also the output response must be carefully checked; it should return either a success (1) or invalid (2) in the case where the input operands are expected to overflow/underflow.

Shift Operations

SHL and SHR need to be tested across a range of inputs and for each port. The specification says that shift operations should not overflow, therefore we should always expect a success (1) response, along with the correct data value.

Other opcodes

The calculator includes a NOP operation; there is not really any output to check / expect for no operation itself, but we must check that after issuing a NOP the calculator is then able to resume processing other instructions successfully.

The `cmd_in` value is 4 bits long, allowing for an additional 11 undefined opcodes to be sent to the calculator. The specification defines the invalid (2) response code as applying to unrecognised commands which we should expect to see in these cases.

Reset

The calculator has a reset facility, it needs to be checked that it does indeed reset all outputs of the calculator, and that after being reset the calculator is then able to function as usual.

Priority Logic

The calculator supports 4 sets of inputs, which the specification says that these can handle 4 requests at once, it should be checked that all 4 can be given inputs at once and return expected results.

The specification also says that they should work using a first come first serve priority algorithm. Therefore we should expect that when all 4 inputs are fed a series of ADD/SUB or SHL/SHR instructions, then each instruction should not complete on a later cycle than any other instructions that begun on the same or a later cycle, otherwise it would violate the priority ordering.