Test cases for DUT - MULT

1. Valid data are given at the input - arg_a, arg_a_parity, arg_b, arg_b_parity.
2. Invalid data are given at the input - parity bits (arg_a_parity, arg_b_parity) are invalid for arg_a and arg_b.
3. Invalid data **a**, valid data **b** are given at the input - arg_b_parity is invalid for arg_b, arg_a_parity is valid for arg_a.
4. Invalid data **b**, valid data **a** are given at the input - arg_a_parity is invalid for arg_a, arg_b_parity is valid for arg_b.
5. ack signal is checked after req set to 1'b1.
6. Check if $result\_rdy$ signal is high after data is converted.
7. Check if data is available for one clock cycle $result\_rdy$.