```
--------------------------------------------------------------------------------
-- Engineer: Olasubomi Borishade
--
-- Create Date: 01/31/2026 09:15:58 PM
-- Design Name: Computation Unit Array Test Bench
-- Module Name: Computation_Array_TestBench - Behavioral
-- Project Name: EENG 5560 Homework 1
-- Description: Testing a 2x8 computational unit (CU) array for 4 bit operands with
4 bit outputs that can compute the following operations:
-- AND, NOR, Addition (ADD), Subtraction (SUB), Multiplication (MULT), Greater than
(GTH), Less than (LTH) & Equal to (EQT)
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
--------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use STD.ENV.FINISH;
use WORK.OPERATIONS_ARRAY_CUSTOMPACK.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Computation_Array_TestBench is
-- Port ( );
end Computation_Array_TestBench;

architecture Behavioral of Computation_Array_TestBench is
Constant d_w_c: integer := 4; -- Declaring data width signal
Constant rows: integer := 2; -- Declaring the number of rows
Constant cols: integer := 8; -- Declaring the number of columns
Signal Ais, Bis, Yis: VectorArray_1d (0 to cols - 1)(d_w_c - 1 downto 0); --
Declaring input and output port signals
Signal DataFlow1is, DataFlow2is: VectorArray_1d (0 to cols - 1)(2 downto 0);
Signal Opselis: OperationArray_2d (0 to rows - 1)(0 to cols - 1); -- Declaring
operation selection port signals

begin
Computation_Unit_Array_Inst: entity work.Computation_Unit_Array(Structural)
    Generic map (d_w => d_w_c,
                rows => rows,
                cols => cols) -- Port-mapping the data width values of the
instatiated computation unit
```

```
    Port map (A => Ais, -- Port-mapping the the input, output and select signals
                B => Bis,
                DataFlow1 => DataFlow1is,
                DataFlow2 => DataFlow2is,
                Opsel => Opselis,
                Y => Yis
                );

stim: process
begin
        Ais <= (("1101", "1100", "1001", "0110", "1010", "1101", "0110", "1100"));
        Bis <= (("0110", "1100", "0011", "0100", "1111", "0111", "0010", "0010"));

        Opselis <= ((aSUB, cEQT, lNOR, aADD, cLTH, lAND, aMULT, cGTH),
                    (NoOp, lNOR, NoOp, aSUB, NoOp, lAND, cGTH, NoOp)); wait for 5ns;

        DataFlow1is <= (("000", "010", "000", "110", "000", "001", "100", "000"));
        DataFlow2is <= (("000", "101", "000", "000", "000", "011", "111", "000"));
wait for 10ns;
finish;
end process;

end Behavioral;
```