ADVANCED DIGITAL DESIGN: EXPERIMENT #6
GENERIC MULTIPLIER/MAC

Purpose

The purpose of this experiment is to write a generic 2^s Complement multiply and accumulate (MAC) unit (for grad students) or multiplier (for undergrads).

References

1) ModelSim Tutorial: from ModelSim, click on Help->PDF Documentation->Tutorial

Problem: Design a Generic Structural Synchronous 2^s Complement MAC (for grad students)

* entity name: *MAC*
* generic constants: *X_len*, *Y_len*: positive, default to 4 (X_len ≤ Y_len); *A_len*: positive, default to 8
* inputs: *X*, *Y*: arbitrary length std_logic_vector (assume ≥ 4); *reset*, *clk*: std_logic
* outputs: *A*: arbitrary length std_logic_vector (assume ≥ X_len+Y_len); *OV*: std_logic

* name your file: MAC.vhd
* include as comments: name and student number (-- precedes a comment line)
* use the array structured multiplication algorithm
* since this is a structural design, only port mapping of components is allowed (e.g., no processes, dataflow equations, functions, procedures, etc.)

* email your main design, components, and testbench(es) to the TA: mousam.hossain@ndsu.edu
* your design will then be run on my testbenches
* make sure that all names and input/output order match those on this sheet, otherwise your design will not run on my testbenches and points will be deducted

* Turn in a report including the following: ***130 total points***
    - project description
    - Block Diagram of MAC Chip
    - gate-level design of MAC components (i.e., FA, HA, AND2, NAND2, OV, DFF0, etc.)
    - component level diagram of the following sized MACs
        > 8+4×4
        > 12+5×4
    - MAC VHDL code
    - VHDL testbench(es), testing 8+4×4 and 12+5×4 MACs

<u>Hints</u>:

- inputs needed to generate OV are: the sign bits of X, Y, Aold, and Anew (Anew = Aold + X * Y)
- the following constructs may be helpful:

```
type twoD_array is array(3 downto 0, 3 downto 0) of std_logic;
signal intermediate: twoD_array;
intermediate(3, 0) <= D;   -- D is defined as std_logic
```

Problem: Design a Generic Structural Boolean 2^s Complement Multiplier (for undergrads)

- entity name: *MULT*
- generic constants: *X_len*, *Y_len*: positive, default to 4 (X_len ≤ Y_len)
- inputs: *X*, *Y*: arbitrary length std_logic_vector (assume ≥ 4);
- outputs: *P*: std_logic_vector of length X_len+Y_len;

- name your file: MULT.vhd
- include as comments: name and student number (-- precedes a comment line)
- use the array structured multiplication algorithm

- email your main design, components, and testbench(es) to the TA: mousam.hossain@ndsu.edu
- your design will then be run on my testbenches
- make sure that all names and input/output order match those on this sheet, otherwise your design will not run on my testbenches and points will be deducted

- Turn in a report including the following: ***130 total points***
  - project description
  - Block Diagram of MULT Chip
  - gate-level design of MULT components (i.e., FA, HA, AND2, NAND2, etc.)
  - component level diagram of the following sized Multipliers
    - > 4×4
    - > 5×4
  - MULT VHDL code
  - VHDL testbench(es), testing 4×4 and 5×4 Multipliers

Hints:

- the following constructs may be helpful:

```
type twoD_array is array(3 downto 0, 3 downto 0) of std_logic;
signal intermediate: twoD_array;
intermediate(3, 0) <= D;   -- D is defined as std_logic
```

- **reports and emailed code due Friday, May 3, but will not be considered late until Tuesday, May 7 at 5:00 pm**