PROJECT: One Dimensional Convolution Engine
Author: Hemil Shah

Part 1

A simple convolution operation between 8-bit input data and 4 bit mask.

1. Design Language: The design has been created as part of the design structure in the System Verilog.

2. Verifying Methodology: The design was thoroughly tested using both the simple testbench and the random testbench. The design passes all the checks in the testbench. Further to this the simple testbench was tweaked a bit to get only one set of inputs (only one convolution to be done) to the design and the waveforms were studied to meet the design expectations. The learning outcome was that it gets difficult to work around issues using the complete module. So, we decided to further divide the Datapath and Control Logic into small modules and instantiate them in the main module. This approach helped us to further simplify our design and debugging became a lot easier. I also think that it might be better to have testbenches for every small module, ensuring correct operations at each module level. This would make things easier to debug and hence get the design working correctly when connected together.

3. Area, Timing and Power: Here are the area, power, maximum clock frequency and critical path's in the logic:

   Area: **1525.509994 um^2**
   Power: **945.5940 uW**
   Max Clock Frequency: **1 GHz**
   Critical Paths: Path from data read from filter memory to being output from accumulator

Design Structure and Description of all the modules

4. **a.** Number of Arithmetic operations: There are 35 arithmetic operations (including multiply and addition as separate computations) needed to convolve vector x of length N=8 and filter vector of length M=4. Generalizing the equation becomes:

   Number of operations = **(N – M + 1) * (2M – 1)**

   This is derived using the fact that the number of convolutions would be (N – M + 1) and for every convolution we perform (2M – 1) operations (2M due to the fact that we do both multiply and accumulate and subtracted by 1 as the number of additions are one less than the number of filters elements).

   b. Module Definitions:
   Control Module: The control module consists of few internal modules:

   i. **Control_f**: This is responsible for generating the ready signal as well as the memory write addresses for the input filter vectors
   ii. **Control_x:** This is responsible for generating the ready signal as well as the memory write addresses for the input data vectors
   iii. **Control_acc:** This module controls the accumulator enable and clear operations
   iv. **Control_mem_rd:** This is responsible for generating the memory address for reading through the memory

   Apart from this the control unit also contains an FSM which controls the current state of the execution. The FSM consists of the following states:

   i. **CT_IDLE:** The control unit is currently IDLE and is waiting for either **s_valid_x** or **s_valid_f** to be asserted (to start the memory write cycle)
   ii. **CT_MEM_WR:** The control unit is currently writing all the input data to X and F memories
   iii. **CT_WAIT:** A wait state before starting the computations

iv. **CT_IN_COMP:** The control unit is processing the convolution for a single output operation

v. **CT_INCR_CYC:** The compute is completed for single output and the number of outputs convoluted till now can be incremented

vi. **CT_DONE:** The data is available at the output port and is waiting for **m_ready_y** to be asserted before again transitioning to IDLE state or compute state (if more vectors are needed to be convolved)

The control unit consists of various counters which help manage the current cycle count for both the memory write operations and the convolution computations. These counters are used to help in state transitions.

c. Verification: The system was verified using the simple testbench. There were various issues solved using the testbench. The simple testbench was modified to input only single convolution values (only one testcase) and the waveforms were dumped to debug any fails. There were failures seen where memory writes were not synced correctly leading the X-propagations and as well as the master ready signals was not correctly being outputted. The simple testbench helped in fixing almost all the issues seen. To our surprise the "Random testbench" passed straight away once all the errors were fixed which were seen using the simple testbench.

d. Number of Cycles: The process takes about 82-cycles to get the entire computation data sent out through the masterbus. The minimum delay in the system comes out to be 82ns (with 1 GHz clock frequency)

e. The area delay product is: 82ns * 1525.509994um^2 = **125091 nsum^2**

f. Energy consumed for one computation: 945.5940uW * 82ns = **77.53 x 10^{-12} J**

g. Number of useful additions and multiplications = **35**. Energy consumed per computation is **22.15x10^{-12} J**

PROJECT: One Dimensional Convolution Engine
Author: Hemil Shah

Part 2:

The design from part 1 is modified top perform a larger convolution with a 128bit input data and 32 bit mask.

Design changes from part 1.

The design written for part 1 is massively parameterized hence to make it work for part 2, there was just an update needed in the parameter's values and the accumulator outputs (to be 21-bits wide). Apart from that the entire Datapath and control logic remains same. Given this we could possibly upscale the design to very large width, but the clock period would also increase with this (as the multiplier would get very slow on increasing the design to larger widths). The control path should continue to work fine with larger widths as well since all the counters and logic scale up accordingly.

1. Area, Timing and Power: Here are the area, power, maximum clock frequency and critical paths in the logic:
   Area: **10855.193885 um^2**
   Power: **8.6574-x10^3 uW**
   Max Clock Frequency: **1 GHz**
   Critical Paths: Path from data read from filter memory to being output from accumulator

   The critical path makes sense absolutely. The path is from the read to Filter memory till the accumulator output. Since the accumulator does intensive computation (multiplication and the addition as well), it makes sense to have the critical path in this data processing area. The control logic is complex but is not data intensive and hence passes the timing checks.

2. Number of cycles: The process takes about **12354-cycles** to get the entire computation data sent out through the master bus. The minimum delay in the system comes out to be **12.354us** (with 1 GHz clock frequency)

   a. The area delay product is: 12.354us * 10855.193885um^2 = **134102.08us um^2**
   b. Energy consumed for one computation: 8.6574-x10^3uW * 12.354us = **106.95 nJ**
   c. Number of useful additions and multiplications = **6111**. Energy consumed per computation is **0.0175 nJ**

PROJECT: One Dimensional Convolution Engine
Author: Hemil Shah

Part 3:
The design form part 2 is optimized to improve the speed

1. To boost the speed, both control unit and Datapath units were doubled up. This was done to allow one unit to read the inputs and store them in memory while the other unit was processing the computations. With this update the computation was outputted at every cycle after the first computation was done thus speeding up the design.

2. Area, Timing, Power: Here is the area, power, clock frequency and

   critical path information: Area: **21732.731776 um^2**
   Power: **1.8121x10^4 uW**
   Max Clock Frequency: **1 GHz**
   Critical Paths: Path from data read from filter memory to being output from accumulator

   The critical path makes sense absolutely. The path is from the read to Filter memory till the accumulator output. Since the accumulator does intensive computation (multiplication and the addition as well), it makes sense to have the critical path in this data processing area. The control logic is complex but is not data intensive and hence passes the timing checks.

3. Number of cycles: The process takes about 12354-cycles to get the entire computation data sent out through the master bus. This is the worst delay possible which occurs for the first convolution. After this the number of cycles is reduced due to multiple control and data-path units. The minimum delay in the system comes out to be **12354ns** (with 1 GHz clock frequency)

   a. The area delay product is: 12354ns * 21732.731776 um^2 = **268477.128usum^2**
   b. Energy consumed for one (worst) computation: 1.8121x10^4 uW * 12.354us = **223.866nJ**
   c. Number of useful additions and multiplications = **6111**. Energy consumed per computation is **0.036634 nJ**