Computer Organization

Homework 3 Report

Name: Rahmet Ali Ölmez

Number: 1801042623

Datapath Design

The datapath gets two 32 bit inputs from the user and control signals from the control circuit. With the commands given from the control circuit, it writes the multiplier to the low part of the product registers. It outputs the least significant bit of the product register and depending on the value, gets a signal of ‘shiftRight’ or ‘AddLeftAndPlace’ from control unit. The datapath sends these two signals to the MathPath block.

The MathPath module outputs the added or shifted value and the new values get written to the registers. After theses operations, the counter iterates and sends a signal indicating whether it has reached 32 or not. Note that what will be written to a register is determined by the signals sent to the multiplexers.

Inside the MathPath module, there is a 64 bit right shifter. This shifter gets the most significant 63 bits of the input and adds a zero to the higher end. This makes the input value shifted one bit right.

Also in MathPath, there is a 32 bit adder. This adder consists of 32 full adders connected to each other. Each full adder gets a total of three inputs and outputs the result and the carry out value. The results of the adders were determined with combinational logic.

Control Unit Design

The control unit is quite simple, it has a register that shows the current state. Depending on the current state and the inputs, the new state is determined. And since the desired output will depend on the current state, the outputs of the control unit are connected to the gates that represent the states. To be able to get the correct state outputs, a state truth table was constructed and the related boolean expressions were implemented (see N2, N1 and N0 blocks in control unit).

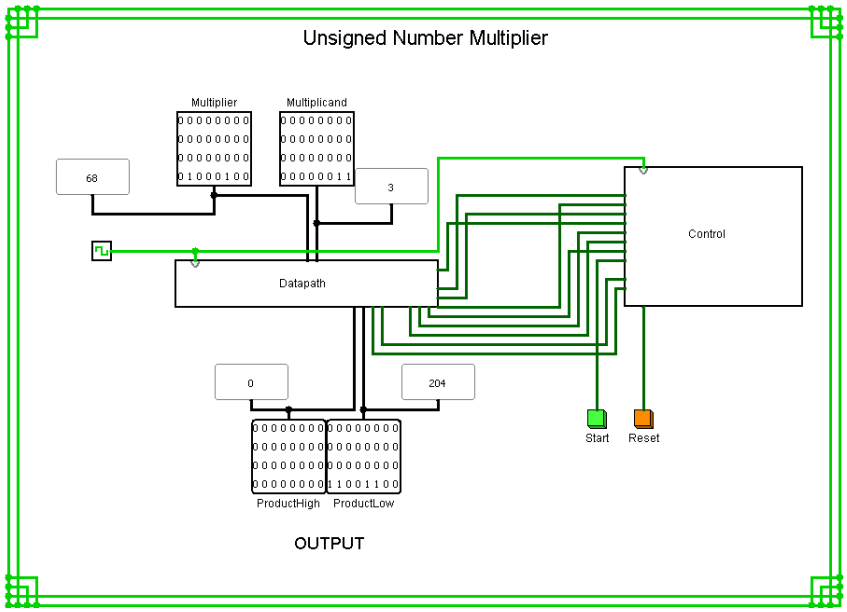
Also, an additional reset signal is sent to the control unit for it to go to the start state and be ready to get inputs again.

Test Cases

Case: Multiply two small numbers: 68 x 3

Expected Output: 204

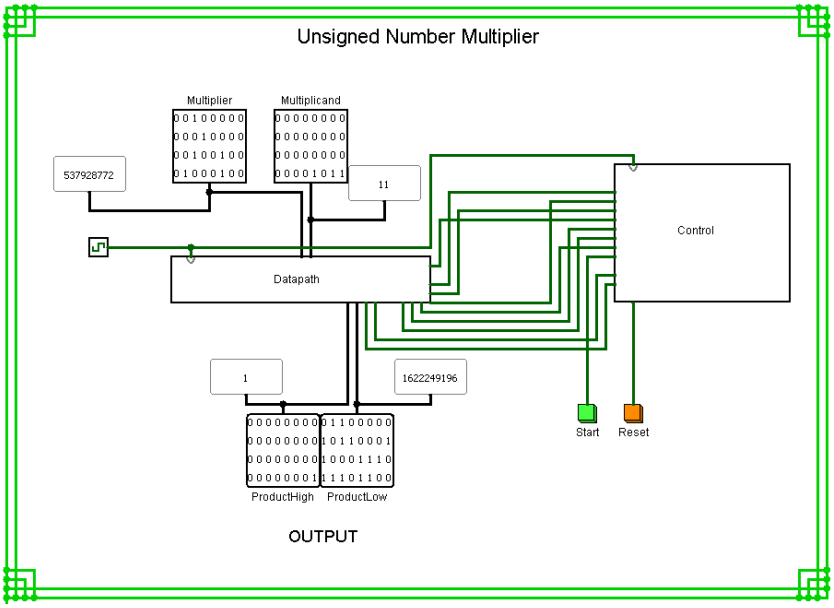
Result: Pass



Case: Multiply one big number and one small number: 537928772 x 11

Expected Output: 5917216492

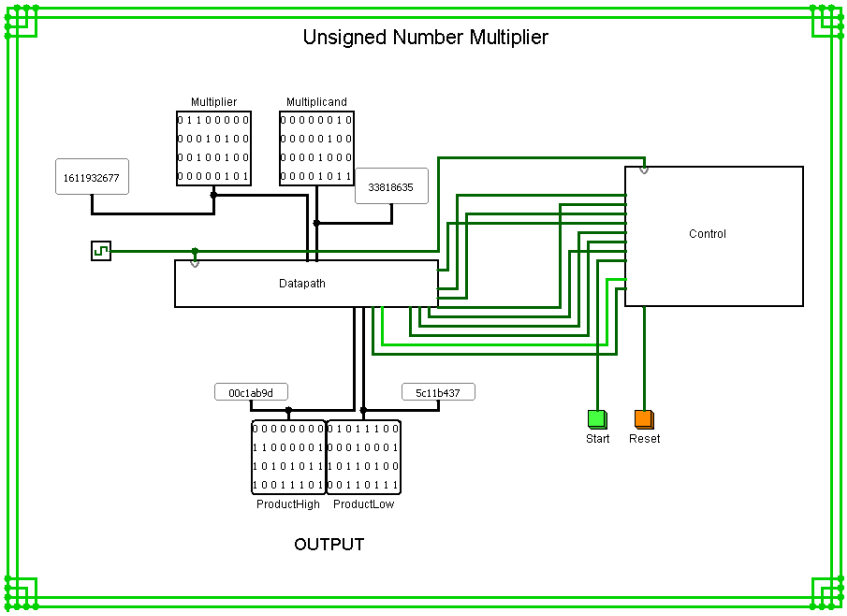
Result: Pass



Case: Multiply two big numbers: 1611932677 x 33818635

Expected Output: 54513362848035895

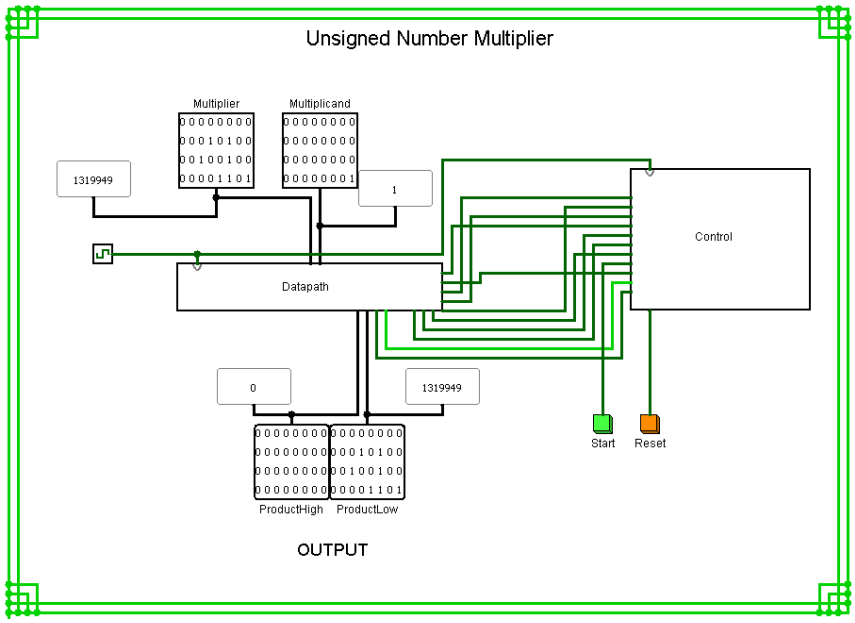
Result: Pass



Case: Multiply with one: 1319949 x 1

Expected Output: 1319949

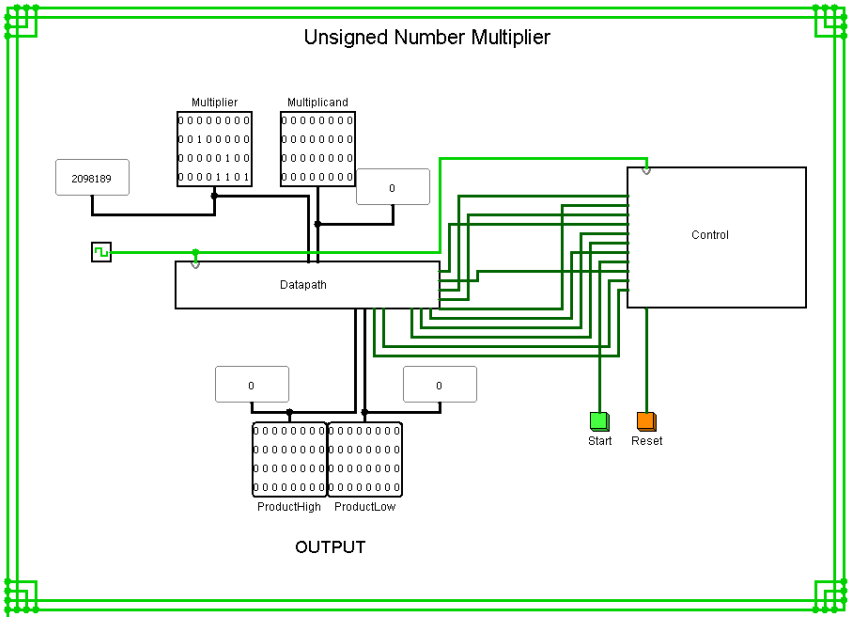
Result: Pass



Case: Multiply with zero: 2098189 x 0

Expected Output: 0

Result: Pass



Case: Multiply with zero (reverse operands): 0 x 6293504

Expected Output: 0

Result: Pass