1.2 Low Latency

A low-latency design is one that passes the data from the input to the output as quickly as possible by minimizing the intermediate processing delays. Oftentimes, a low-latency design will require parallelisms, removal of pipelining, and logical short cuts that may reduce the throughput or the max clock speed in a design.

Referring to *iterative_power3.sv*, there is no obvious latency optimization to be made to the as each successive multiply operation must be registered for the next operation. However, the registers in the pipelined implementation in *pipelined_power3.sv* can be stripped out to reduce the input to output timing.

This has been implemented in *combinational_power3.sv*, where each stage is a combinatorial expression of the previous stage as shown in Figure 1.2.

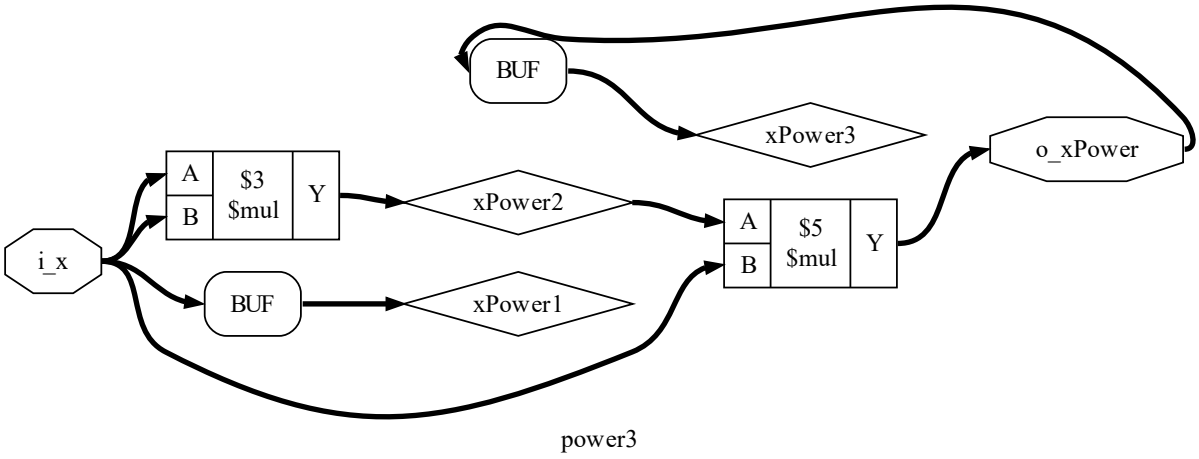

Figure 1.2: Synthesized design of a combinational implementation to calculate the cube of a number.

The performance of this design is:

- Throughput = 8 bits/clock (assuming one new input per clock)
- Latency = Between one and two multiplier delays, 0 clocks
- Timing = Two multiplier delays in the critical path

By removing the pipeline registers, the latency of this design has been reduced to less than a single clock cycle. The penalty is in the timing. The implementations in Section 1.1, could theoretically run the system clock period close to the delay of a single multiplier, but in this implementation, the clock period must be at least two multiplier delays plus any external logic in the critical path.