2.1 Rolling up the pipeline

One way to minimize the area of a pipelined design is to roll up the pipeline so that logic can be reused.

Consider the fixed-point fractional multiplier implemented in *mult8.sv*, which takes one clock cycle to generate the product. The multiplier would be synthesized into a long chain of logic that can be easily pipelined by adding intermediate register layers. An alternative approach is to perform the multiplication using a series of shift and add operations, as implemented in *shift_add_mult8.sv*. This approach results in a more compact multiplier design as shown in *shift_add_mult8.sv*g, but it will now require 8 clocks to complete a multiplication.

Note: the synthesized design shows more registers than expected as Yosys optimizes the design, by converting the shift operations into registers. When the design gets mapped to a generic library, Yosys replaces these registers with wires as expected. To demonstrate this, *test.sv* implements a register that gets left shift at each clock cycle and its synthesized design mapped to a generic library is shown Figure 2.1.

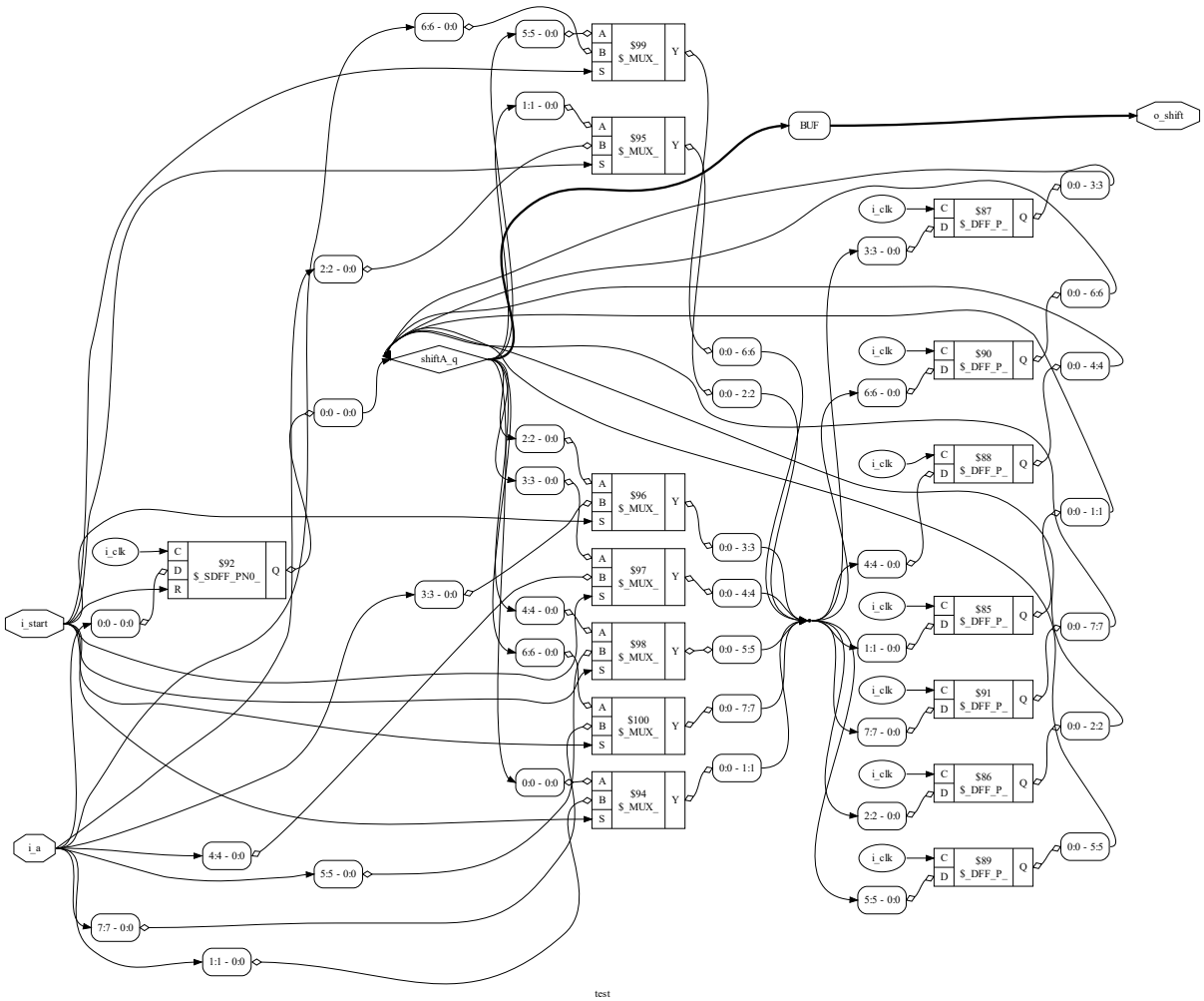

Figure 2.1: Synthesized design of test.sv mapped to a generic library of built-in logic gates and registers.