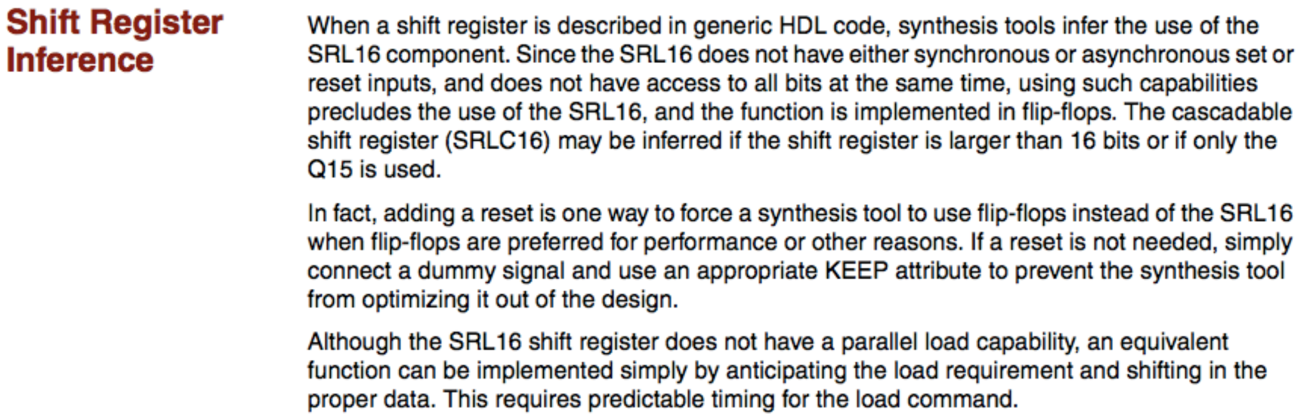
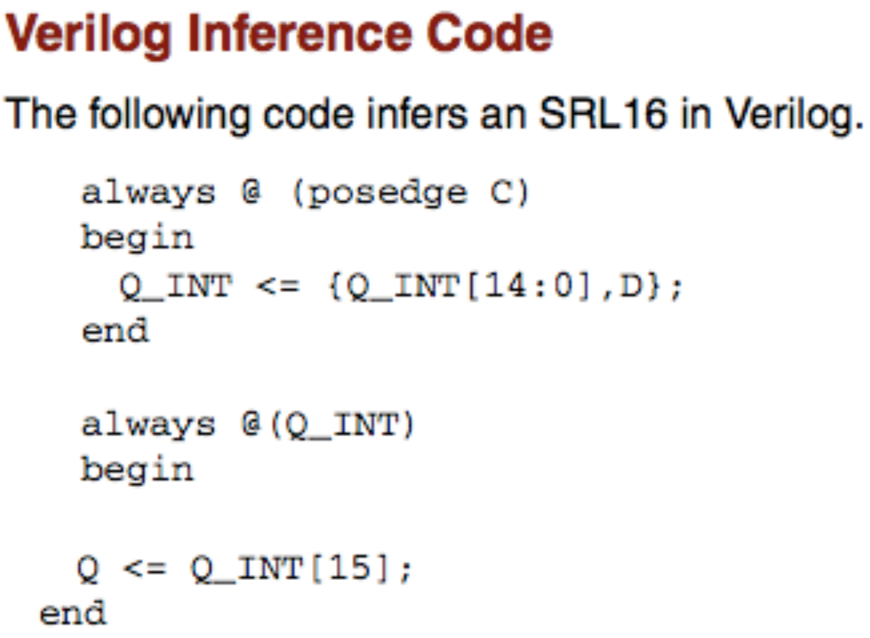


Shift Register Inference

When a shift register is described in generic HDL code, synthesis tools infer the use of the SRL16 component. Since the SRL16 does not have either synchronous or asynchronous set or reset inputs, and does not have access to all bits at the same time, using such capabilities precludes the use of the SRL16, and the function is implemented in flip-flops. The cascadable shift register (SRLC16) may be inferred if the shift register is larger than 16 bits or if only the Q15 is used.

In fact, adding a reset is one way to force a synthesis tool to use flip-flops instead of the SRL16 when flip-flops are preferred for performance or other reasons. If a reset is not needed, simply connect a dummy signal and use an appropriate KEEP attribute to prevent the synthesis tool from optimizing it out of the design.

Although the SRL16 shift register does not have a parallel load capability, an equivalent function can be implemented simply by anticipating the load requirement and shifting in the proper data. This requires predictable timing for the load command.



Verilog Inference Code

The following code infers an SRL16 in Verilog.

```
  always @ (posedge C)
  begin
    Q_INT <= {Q_INT[14:0],D};
  end

  always @(Q_INT)
  begin

  Q <= Q_INT[15];
end
```

NOTE: The 2nd always statement is needed because Q[15] is the output of the SRL.
“Q_INT” is only a temporary variable.