

Barrel shifter Although Verilog has built-in shift functions, there is no rotation operation. In this sub-section, we examine an 8-bit barrel shifter that rotates an arbitrary number of bits to the right. The circuit has an 8-bit data input, *a*, and a 3-bit control signal, *amt*, which specifies the amount to be rotated. The first design uses a case statement to exhaustively list all combinations of the *amt* signal and the corresponding rotated results.

The code is straightforward, it will become cumbersome when the number of data bits increases. Furthermore, a large number of items in a case statement implies a wide multiplexer, which makes synthesis difficult and leads to a large propagation delay. Alternatively, we can construct the circuit by stages. In the *n*th stage, the input signal is either passed directly to output or rotated right by 2^{*n*} positions. The *n*th stage is controlled by the *n*th bit of the *amt* signal. Assume that the 3 bits of *amt* are *m2m1m0*. The total rotated amount after three stages is $m_22^2 + m_12^1 + m_02^0$, which is the desired rotating amount