Verilog Coding Style ala Blue Chip Designs, Inc.

The three (3) most basic rules to produce better and more reusable, designs are:

1. Use a fully synchronous design style, and most importantly register the inputs and outputs of macros or "cores". This makes timing optimization a local problem.

2. Do rigorous, bottom up verification; make sure a block or module is completely verified before it is integrated into the next level of hierarchy. This helps to make verification as local as possible.

3. Plan before doing; write a reasonable specification before design begins. This initial planning can help produce well-architected, well-partioned designs that can employ locality effectively.

Applying these 3 rules, especially **registering the inputs and outputs**, can have dramatic effect in reducing design time.

Can make designs reusable. Simply put, without employing these rules, designing large chips will not be possible.

To be reusable, the design must first be usable: a robust and correct design.
1. good documentation
2. good code
3. thorough commenting
4. well-designed verification environments and suites
5. robust scripts

➔ Initially, complying with these design reuse practices might seem like an extra burden, but once the design team is fully trained, these techniques speed the design, verification and debug processes of a project by reducing iterations throughout the code and verification loop.

Design for reuse:
1. Designed to solve a general problem
2. Designed for use in multiple technologies
3. Designed for simulation with a variety of simulators
4. verified independently of the chip in which it will be used
5. verified to a high level of confidence
6. fully documented in terms of appropriate applications and restrictions

Verilog signals/variables/identifiers and coding style:

1. All modules must have a brief description of its functions as well as description of the signals used in the module.
2. There must be a modification history for each module. Example of guides 1 & 2:

```
/*
-- This confidential and proprietary software was design by Blue Chip Designs, Inc.
-- as requested by Cyber Imaging, Inc.
--
-- File     : ad5308.v
-- Date     : Nov 29, 2005
-- Version  : 1.0
-- Abstract : This controls the ad5308 IC.  This module generates the necessary
           signals to operate DAC ad5308 including SCLK, SYNC, DIN and LDAC.
-- Signal Description
        input signals
           RST_n     = master reset signal
           CLK       = master clock signal, at 20.8 ns period
           DATA      = bit 15 (MSB) to bit 0 (LSB) respectively are:
                   =  D'/C, A2, A1, A0, D7, D6, D5, D4, D3, D2, D1, D0, 0, 0, 0, 0
                     D'/C = 0 means DAC write, 1 means control fcn so defaulted to 0
                     A2A1A0 = DAC channel
                   = 0,A2,A1,A0,D7,D6,D5,D4,D3,D2,D1,D0,0,0,0,0
           START     = a positive pulse signal to signify start of DAC conversion
--         SCLK      = serial clock that controls DIN
--         SYNC_n    = low during SCLK
--         DIN       = serial output of DATA starting from bit 15 (MSB)
--         LDAC_n    = end of conversion, trasnfer of output to specified channel.
--
-- Modification History:
-- Date         By       Version      Change Description
-- 10/25/05    cmo       1.0          Original
-- 11/29/05    cmo       1.1          SCLK null state converted to 1 from 0
-- 11/30/05    cmo       1.2          masked DATA = DATAin & 16'h7ff0
  ============================================================
*/
```

3. Each module must have a reasonable testbench.
4. Use all capital letters for port identifiers.
5. Exception to number 4, use of numbers, use of "n" to signify active low signals and use of "p" to indicate number of pulses, use of "d" to indicate delayed version of a signal
   Example CLK, RST_n, CLK_4p, CLK_n3p
   RST_n is an active-low signal, resets when the signal is low.
   CLK_4p is an active-high signal with pulse width of 4 clock cycles.
   CLK_n3p is an active-low signal with a low pulse width of 3 clock cycles.
   CLK_2d is a two clock period delayed version of CLK

When in doubt, ask your supervisor ☺