Area Improve：

1. Shift Register：Using bitwise shifts to replace multiplication and division by powers of two can save a lot of resources. Note: <<< or << (with or without sign extension)

   When using shifts to replace division, ensure that the higher bits are sufficient.
   For example:

   ```
   reg [3:0] a;    a >> 5;
   ```

   If 'a' needs all 4 bits, this will cause issues. It can be fixed as:

   ```
   reg [8:0] a;    a >> 5;
   ```

2. Shared registers.

3. Take off unused reset.

4. Signle bit flag：
We can write reg[8] == 1 instead of reg == 256.

Escape Timing Violation：

1. Some combinational circuits are too long and can span across a register to avoid not completing within one clock cycle, which can cause timing violations.

2. At the macro interfaces, try to use registers to prevent timing violations during APR.

3. In the if conditions, try to use the earliest driven register (such as in_valid); otherwise, timing violations might occur at the gate level.

4. If a multiplication or division line is too long, we can use multi-stage multiplication or division to fix it.