**PROBLEM STATEMENT:**

Implement **Double dabble** or **shift and add-3** algorithm to convert a binary number into Binary-coded Decimal.  Convert the output Binary coded Decimal to Binary number using Reverse Double Dabble. Register Numbers, Shift flow, and speed of execution should be tested with 5 samples.

**EXPLANATION:**

convert binary numbers to decimal numbers by reading them from left to right, starting with a decimal value of 0, iterating over the digits of the binary number, adding 1 to the decimal number for every 1 you reach in the binary number, and multiplying by 2, when you move on to the next digit. we have to add 3, when a number in a BCD-field exceeds 4 before shifting.

**ALGORITHM:**

Step – 1 Divide the binary number which is to be converted by two which is the base of the decimal number.

Step – 2 The remainder which is obtained from step 1 is the least significant bit of the new decimal number.

Step – 3 Divide the quotient which is obtained from the step 2 and the remainder obtained from this is the second least significant bit of the decimal number.

Step – 4 Repeat the process until the quotient remains zero.

Step – 5 The last remainder obtained from the division is the most significant bit of the decimal number. Hence arrange the number from most significant bit to the least significant bit (i.e., from bottom to top).

**LOGIC :**

The above logic is a single digit Binary to BCD converter, contained in the binary\_to\_bcd\_digit. The higher level binary\_to\_bcd file instantiates one of these single digit converters for each digit of the BCD output and cascades them together to form a multi-digit Binary to BCD converter. This higher level logic also latches in the initial binary value from the user logic and stores it in a shift register. It controls the enables of the single digit converters and executes the appropriate number of shifts into it from the binary shift register. Finally, it outputs the result.