

Division

- How do we perform division on binary numbers?

$$\begin{array}{r} \text{Divisor } 10 \) \ 111 \\ \underline{-10} \\ 11 \\ \underline{-10} \\ 1 \end{array} \quad \begin{array}{l} \leftarrow \text{Quotient} \\ \leftarrow \text{Dividend} \\ \\ \\ \leftarrow \text{Remainder} \end{array}$$

$$\text{Dividend} = \text{Quotient} \times \text{Divisor} + \text{Remainder}$$

Dividend and Remainder should have the same sign.

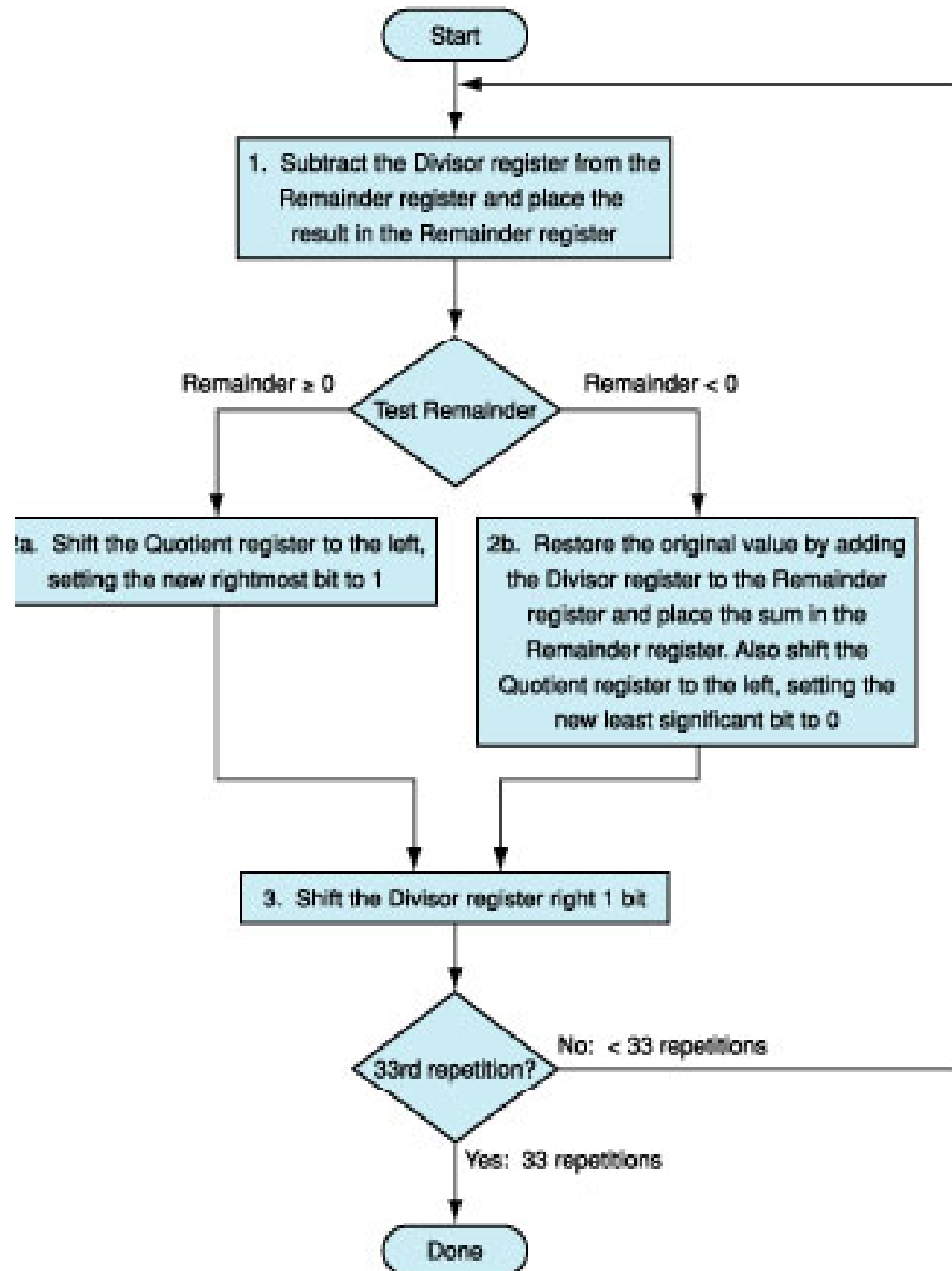
Control for Division

Remainder register is initialized to the Dividend value.

Divisor is shifted to left so that the left most digit is aligned with the left most digit of the dividend.

To decide whether divisor can be subtracted from remainder, we need to subtract and check if the result is negative or positive.

If we cannot subtract, we need to add the subtracted value back to the remainder.



Example: Divide 0000 0111 by 0010

Iteration	Step	Quotient	Divisor	Remainder
0	Initial value	0000	0010 0000 (sll to be 8 bit)	0000 0111 (=Dividend)
1	1. Rem=Rem-Div 2b. Rem<0, Rem+=Div, sll Q, Q0=0 3. srl Div	0000 0000 0000	0010 0000 0010 0000 0001 0000	1110 0111 0000 0111 0000 0111
2	1. Rem=Rem-Div 2b. Rem<0, Rem+=Div, sll Q, Q0=0 3. srl Div	0000 0000 0000	0001 0000 0001 0000 0000 1000	1111 0111 0000 0111 0000 0111
3	1. Rem=Rem-Div 2b. Rem<0, Rem+=Div, sll Q, Q0=0 3. srl Div	0000 0000 0000	0000 1000 0000 1000 0000 0100	1111 1111 0000 0111 0000 0111
4	1. Rem=Rem-Div 2a. Rem>=0, sll Q, Q0=1 3. srl Div	0000 0001 0001	0000 0100 0000 0100 0000 0010	0000 0011 0000 0011 0000 0011
5	1. Rem=Rem-Div 2a. Rem>=0, sll Q, Q0=1 3. srl Div	0001 0011 0011	0000 0010 0000 0010 0000 0001	0000 0001 0000 0001 0000 0001

- We need to move the divisor to the right one digit each time, so we start with the divisor place in the left half of the 64-bit Divisor register and shift it right to w bit each step to align it with the dividend. Divisor and Remainder registers are 64-bit long, and we also need to use 64-bit ALU.

