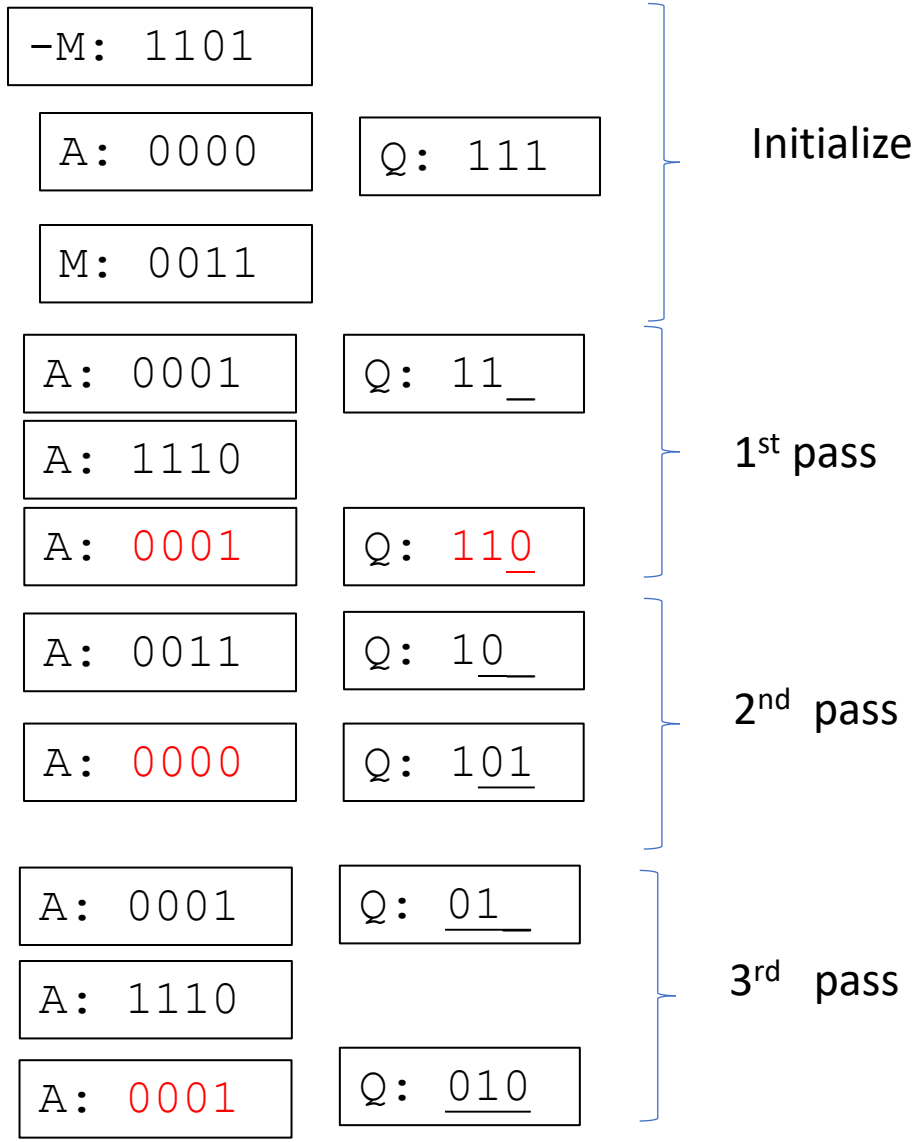
Restoring Method

- In *restoring division*, the multiplier to the divisor is determined by first subtracting the divisor. If the result is negative, the divisor is restored (added back).

Restoring Method

- Initialization
  - Clear *A*. Requires 1 extra bit for *A* to be used as a sign bit.
  - *Q* gets dividend.
  - *M* gets divisor.
- Loop for each bit of the dividend *Q*
  - Shift *AQ* to the left.
  - Subtract ($A \leftarrow A - M$)
  - If negative ($A_n = 1$), restore ($A \leftarrow A + M$) and reset Q_0 ($Q_0 \leftarrow 0$)
  - Else set Q_0 ($Q_0 \leftarrow 1$)
- Quotient in *Q* while remainder in *A*; adjust sign as needed

Restoring Division



111 (Q) / 11 (M)

- Initialization
  - Clear *A*. Requires 1 extra bit for *A* to be used as a sign bit.
  - *Q* gets dividend.
  - *M* gets divisor.
- Loop for each bit of the dividend *Q*
  - Shift *AQ* to the left.
  - Subtract ($A \leftarrow A - M$)
  - If negative ($A_n = 1$), restore ($A \leftarrow A + M$) and reset Q_0 ($Q_0 \leftarrow 0$)
  - Else set Q_0 ($Q_0 \leftarrow 1$)
- Quotient in *Q* while remainder in *A*; adjust sign as needed

Try

Try: 01101 (Q) / 00101 (M) (using restoring division)
Show the value of A and Q after the end of each pass

After this pass	A	Q
1st	000000	11010
2nd	000001	10100
3rd	000011	01000
4th	000001	10001
5th	000011	00010