

CSARCH Lecture Series: Restoring Division

Sensei RL Uy

College of Computer Studies

De La Salle University

Manila, Philippines



Copyright Notice

This lecture contains copyrighted materials and is use solely for instructional purposes only, and not for redistribution.

Do not edit, alter, transform, republish or distribute the contents without obtaining express written permission from the author.

Overview

Reflect on the following question:

- How does Arithmetic and Logic Unit (ALU) perform division?

```
int main()
{
    int var, var1, var2;
    var = 5;
    var1 = 2;
    var2 = var / var1;
}
```



Overview

- This sub-module describes how Arithmetic Logic Unit (ALU) performs division using restoring division
- The objective is as follows:
 - ✓ Describe the process of performing restoring division

Integer Division

- Unlike subtraction, division cannot easily be performed using multiplication since it is difficult to take the reciprocal of a number.
- Division is performed by repeated subtraction
- There are two methods for integer binary division
 - Restoring
 - Non-restoring

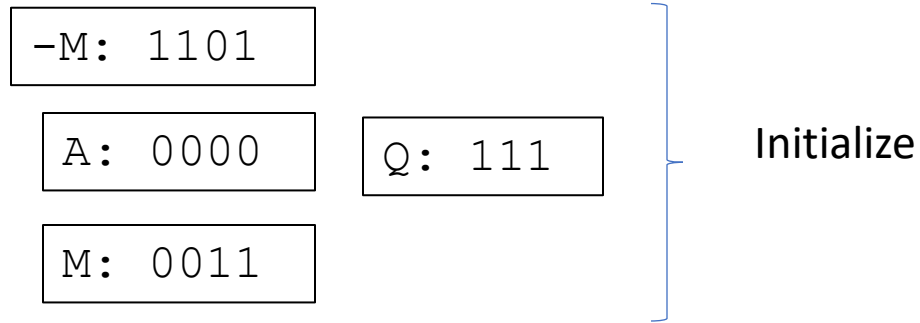
Restoring Method

- 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

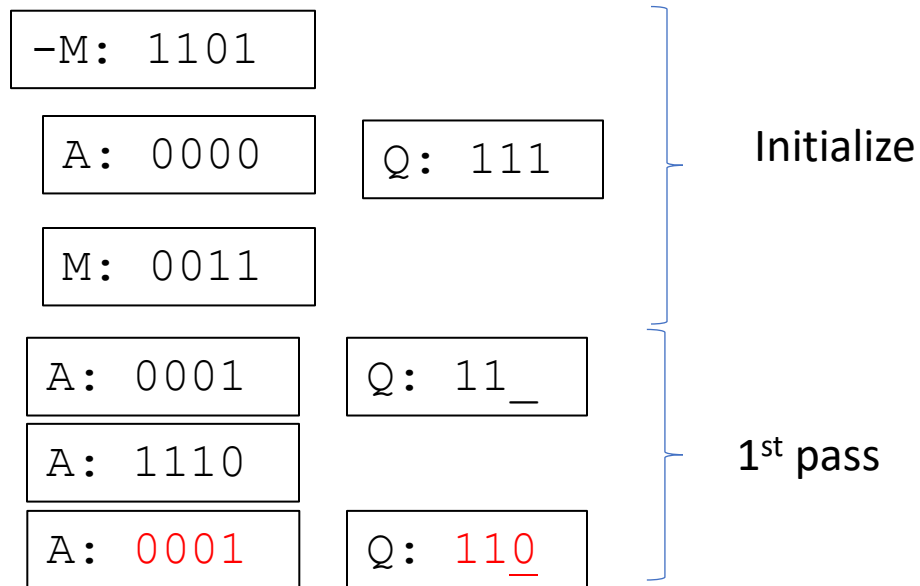
Restoring Division



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

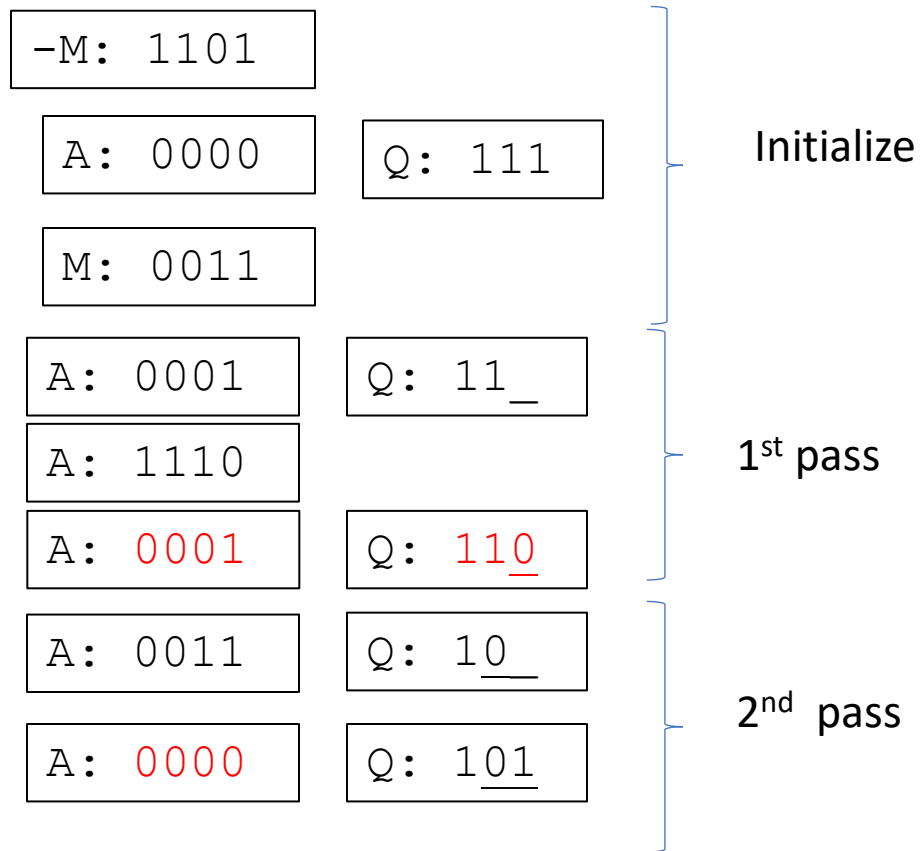
Restoring Division



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

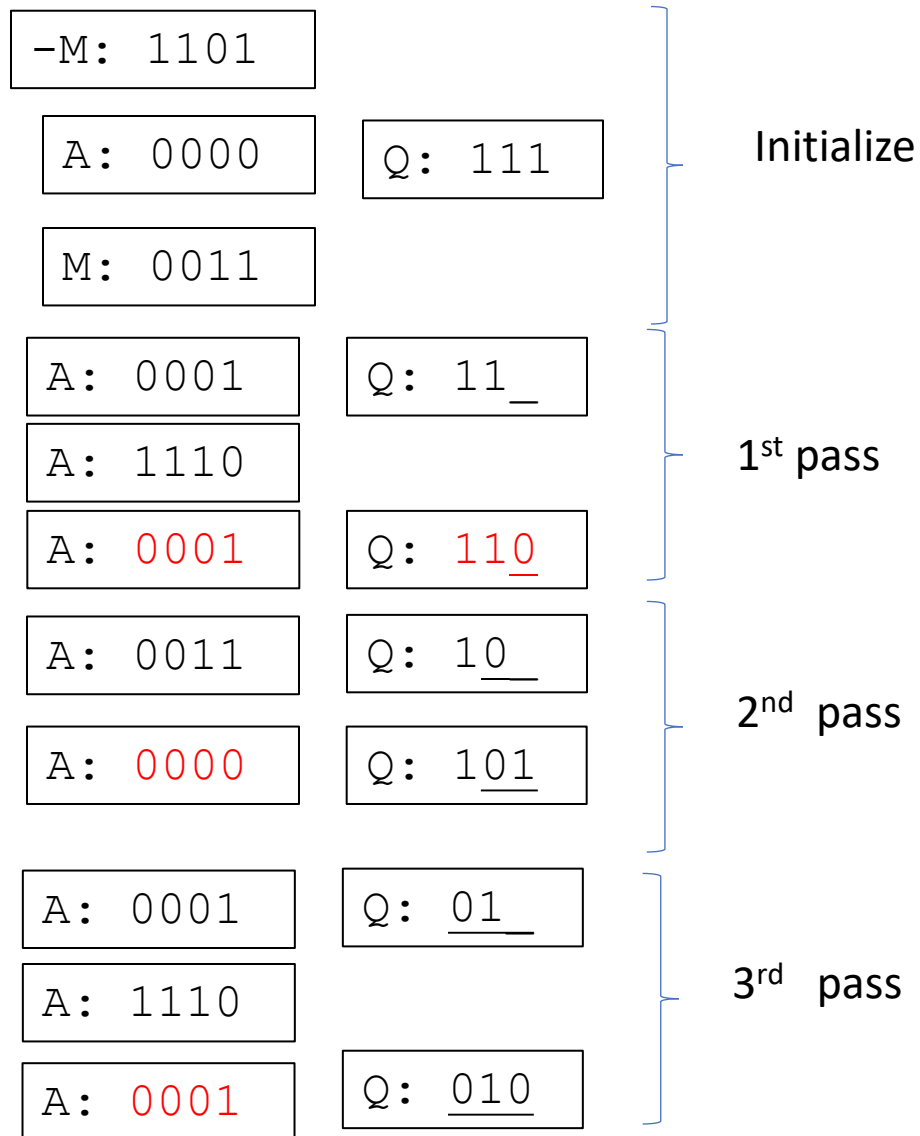
Restoring Division



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

Restoring Division



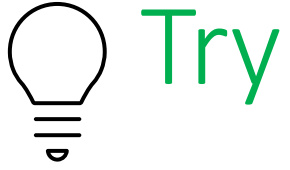
111 (Q) / 11 (M)

- 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: 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 |
|-----------------|---|---|
| 1 st | | |
| 2 nd | | |
| 3 rd | | |
| 4 th | | |
| 5 th | | |



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 |
|-----------------|--------|-------|
| 1 st | 000000 | 11010 |
| 2 nd | 000001 | 10100 |
| 3 rd | 000011 | 01000 |
| 4 th | 000001 | 10001 |
| 5 th | 000011 | 00010 |

To recall ...

- What have we learned:
 - ✓ Describe the process of performing restoring division