Outline of Lecture

- **Division Algorithms**
- **Division Hardware**

Division

- The final arithmetic operation to be included in the ALU is ***division***.

- Paper-and-pencil example:

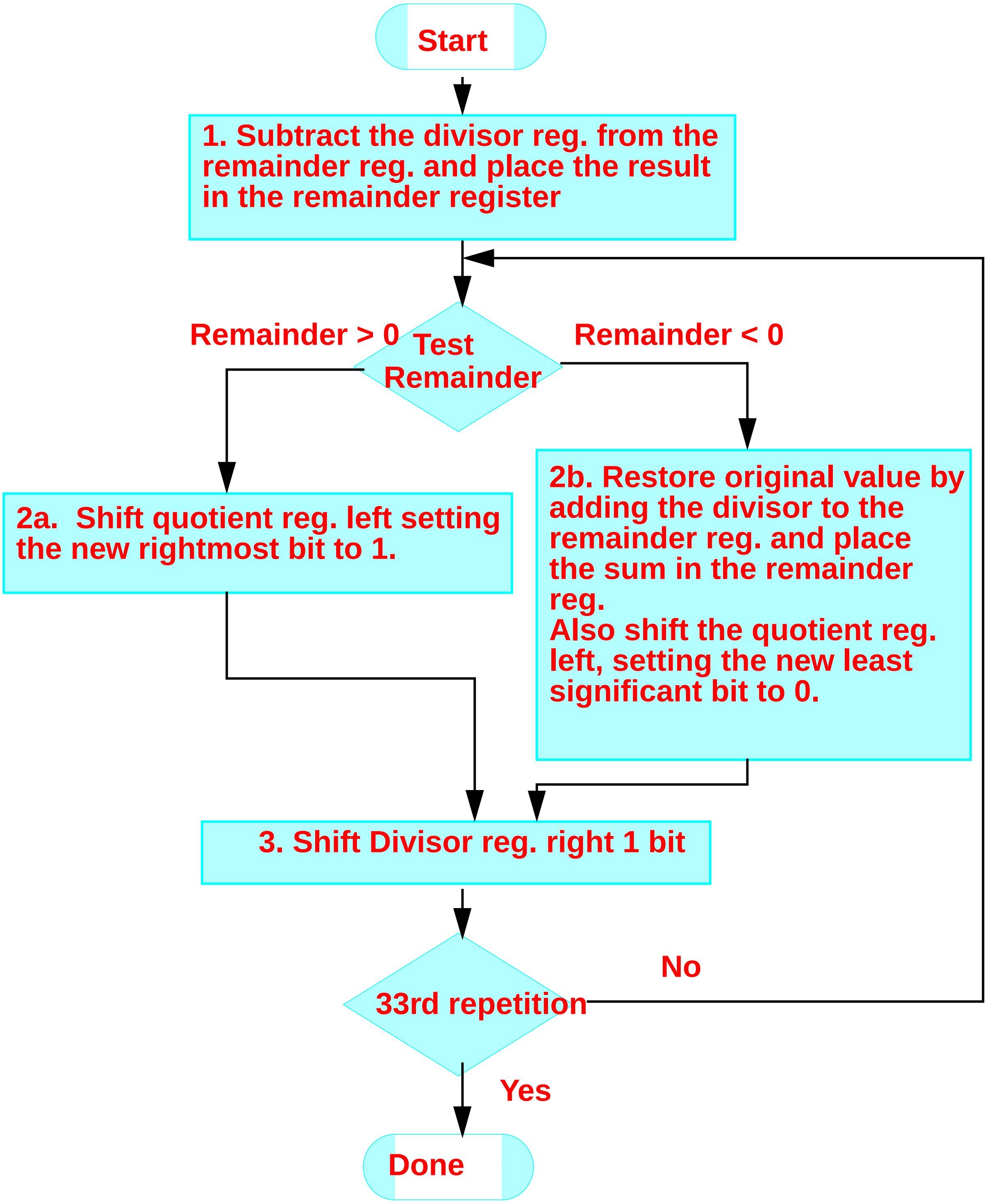
```
                   1001  Quotient
Divisor 1000    1001010  Dividend
               -1000
                   10
                   101
                   1010
                  -1000
                     10  Remainder
```

Dividend = Quotient x Divisor + Remainder

- See how big a number can be subtracted, creating quotient bit on each step:

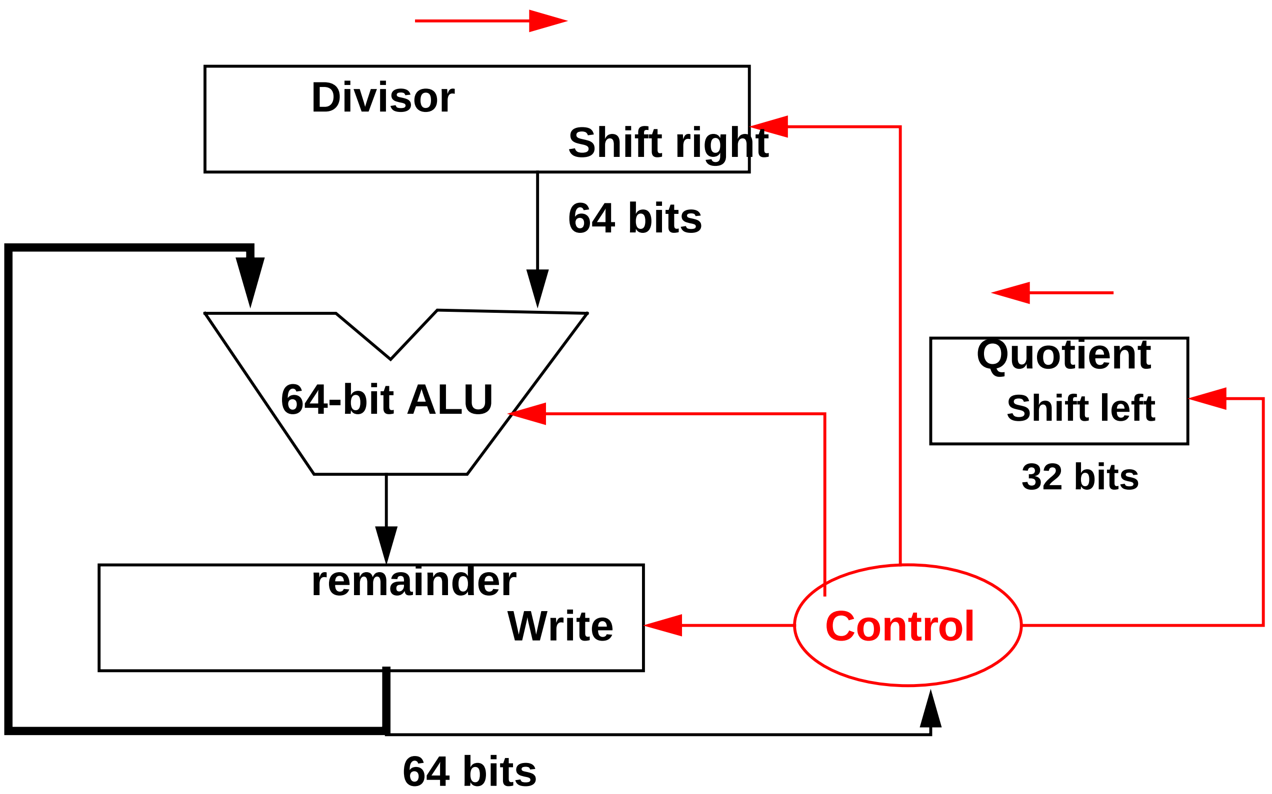
- Binary $\Rightarrow$ 1 x divisor or 0 x divisor

First Version



Division Hardware

- The hardware needed to implement the paper-and-pencil algorithm is as follows:



Example

Using 4-bit numbers, divide 0000 0111_2 by 0010_2.

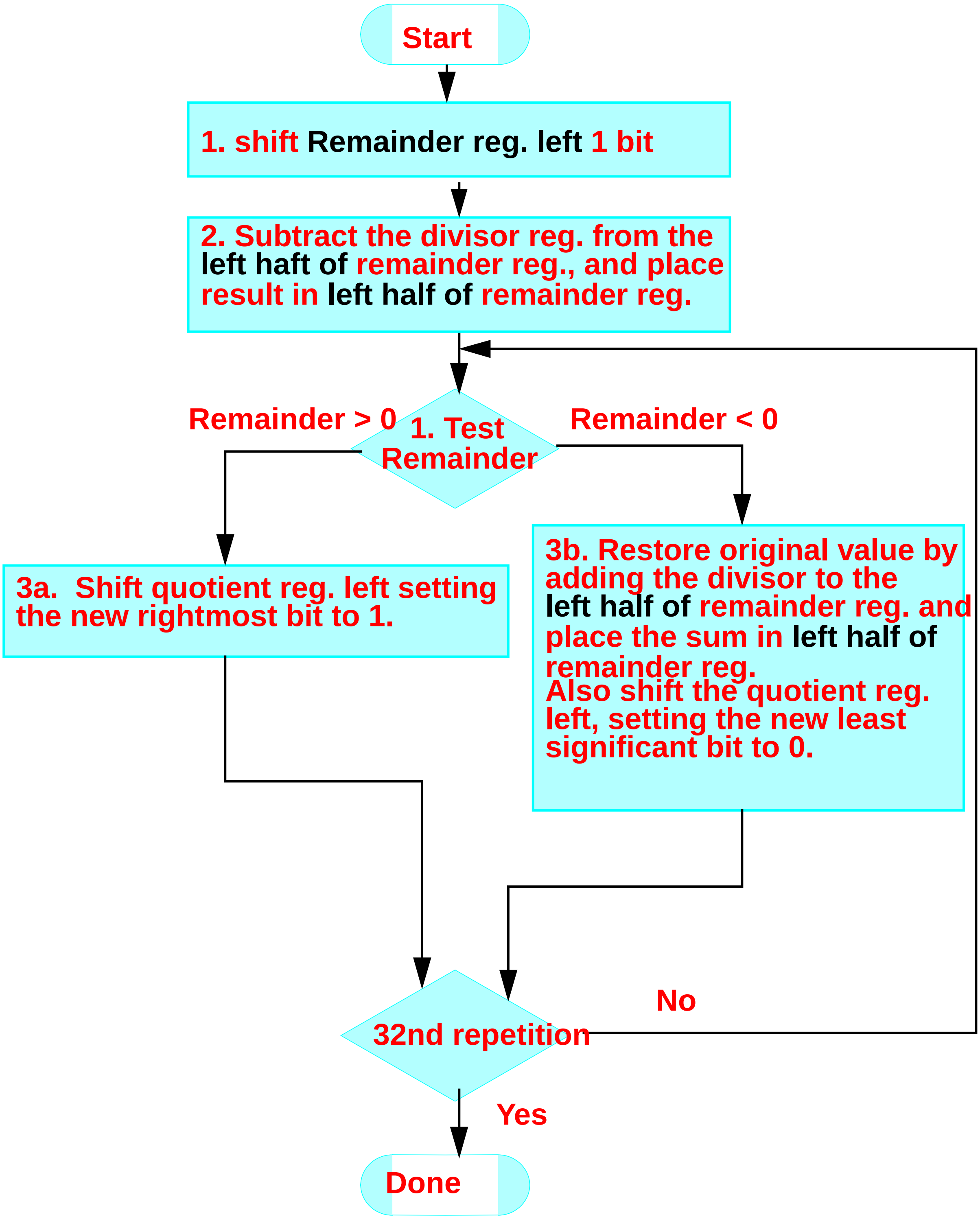
Solution

Iteration	Step	Quotient	Divisor	Remainder
0	Initial Values	0000	0010 0000	0000 0111
1	1a: Rem = Rem - Div	0000	0010 0000	1110 0111
	2b: Rem<0 ⇒ +Div, sll Q, Q0=0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1a: Rem = Rem - Div	0000	0001 0000	1111 0111
	2b: Rem<0 ⇒ +Div, sll Q, Q0=0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111
3	1a: Rem = Rem - Div	0000	0000 1000	1111 1111
	2b: Rem<0 ⇒ +Div, sll Q, Q0=0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111
4	1a: Rem = Rem - Div	0000	0000 0100	0000 0011
	2b: Rem>0 ⇒ sll Q, Q0=1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011
5	1a: Rem = Rem - Div	0001	0000 0010	0000 0001
	2b: Rem>0 ⇒ sll Q, Q0=1	0011	0000 0010	0000 0001
	3: Shift Div right	0011	0000 0001	0000 0001

Observations

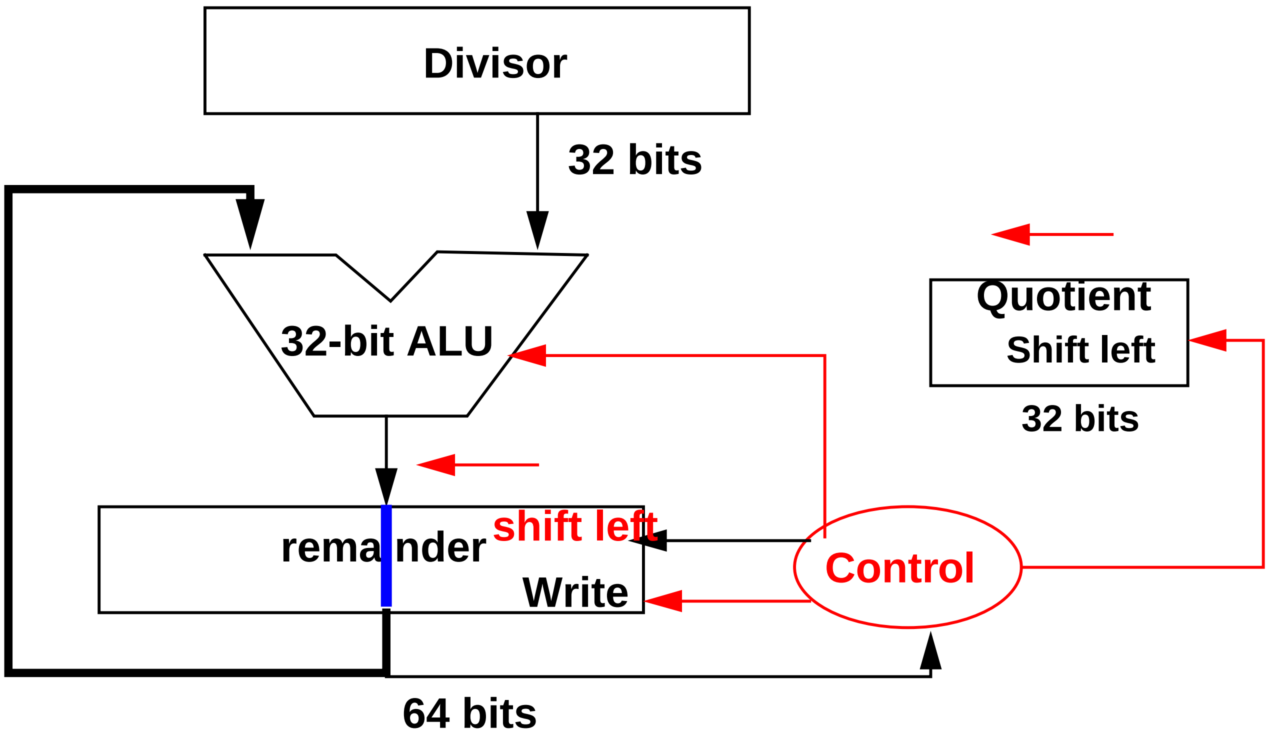
- The division needs a 64-bit ALU. However, only half of the divisor bits contain useful information.
- On the other hand, shifting the remainder to the left instead of shifting the divisor to the right produces the same alignment and accomplishes the same goal.
- Thus, we could limit the size of the divisor to the 32 bits, and in this case, we will need just a 32-bit ALU to accomplish division - which can be done by our 32-bit MIPS ALU.

Second Version



Hardware

- The hardware needed to implement this new version of the division algorithm is as follows:



Example

Using 4-bit numbers, divide 0000 0111$_2$ by 0010$_2$.

Solution

Iteration	Step	Quotient	Divisor	Remainder
0	Initial Values	0000	0010	0000 0111
1	1: Shift Remainder left	0000	0010	0000 1110
	2: Rem = Rem - Div	0000	0010	1110 1110
	3b: Rem < 0 ⇒ + Div, sll Q, Q0 = 0	0000	0010	0000 1110
2	1: Shift Remainder left	0000	0010	0001 1100
	2: Rem = Rem - Div	0000	0010	1111 1100
	3b: Rem < 0 ⇒ + Div, sll Q, Q0 = 0	0000	0010	0001 1100
3	1: Shift Remainder left	0000	0010	0011 1000
	2: Rem = Rem - Div	0000	0010	0001 1000
	3a: Rem > 0 ⇒ sll Q, Q0 = 1	0001	0010	0001 1000
4	1: Shift Remainder left	0001	0010	0011 0000
	2: Rem = Rem - Div	0001	0010	0001 0000
	3a: Rem > 0 ⇒ sll Q, Q0 = 1	0011	0010	0001 0000