RISC V Architecture

Prof. H R Vanamala

Department of Electronics and Communication Engg.

PES UNIVERSITY
CELEBRATING 50 YEARS

RISC V ARCHITECTURE

UNIT 4: Arithmetic for Computers

Prof. H R Vanamala

Department of Electronics and Communication Engineering

Division:

Similar to long division using decimal numbers.

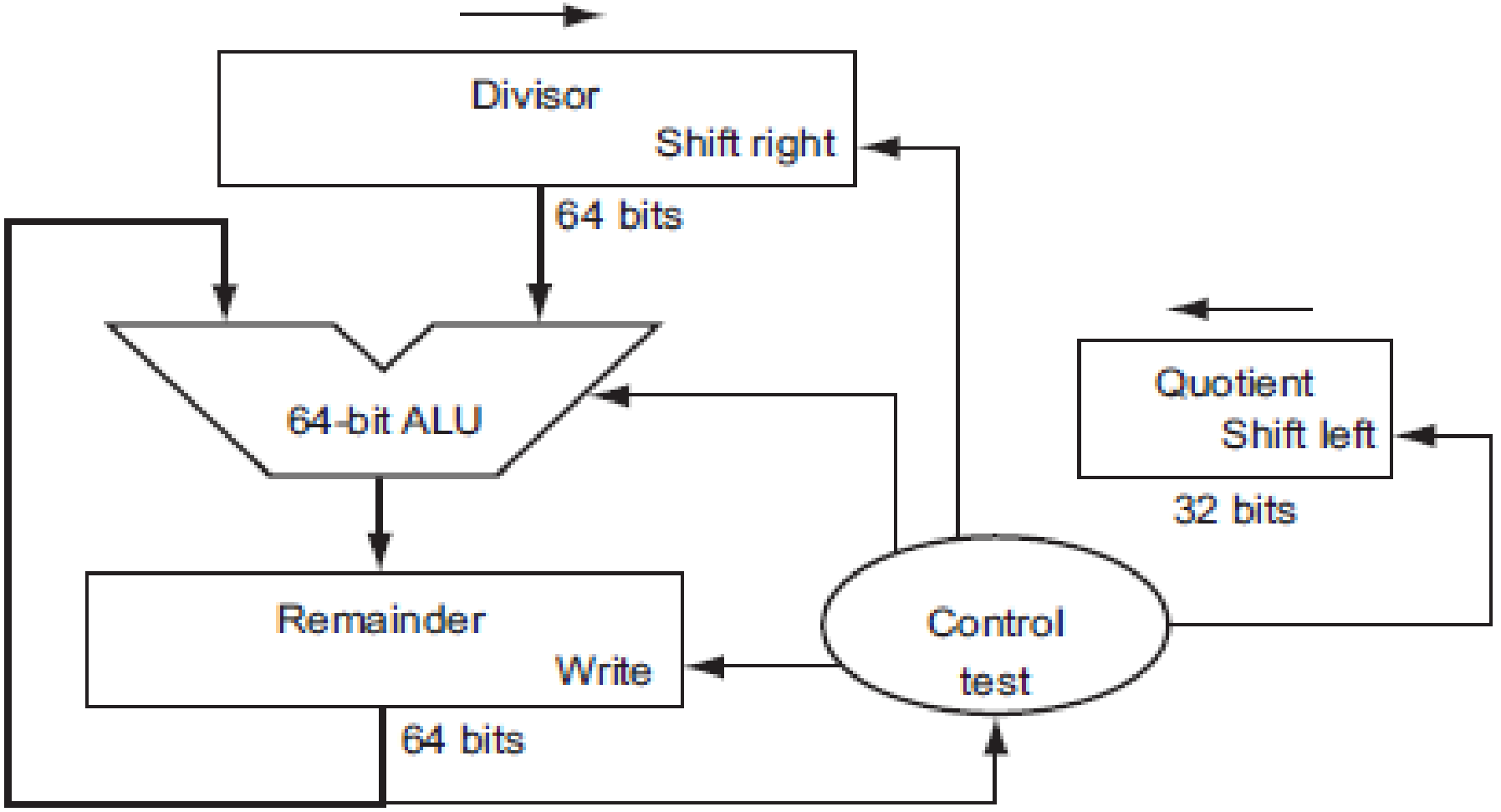
Consider the case with both the dividend and the divisor positive and hence the quotient and the remainder are nonnegative.

Example: Divide $1{,}001{,}010_{ten}$ by 1000_{ten}:

Dividend = Quotient × Divisor + Remainder

$$
\begin{array}{rll}
 & 1001_{ten} & \text{Quotient} \\
\text{Divisor } 1000_{ten} \;\big|\; & 1001010_{ten} & \text{Dividend} \\
 & -1000 & \\
\hline
 & 10 & \\
 & 101 & \\
 & 1010 & \\
 & -1000 & \\
\hline
 & 10_{ten} & \text{Remainder}
\end{array}
$$

A Division Hardware:



A Division Algorithm:

•The Divisor register, ALU, and Remainder register are all 64 bits wide, with only the Quotient register being 32 bits.

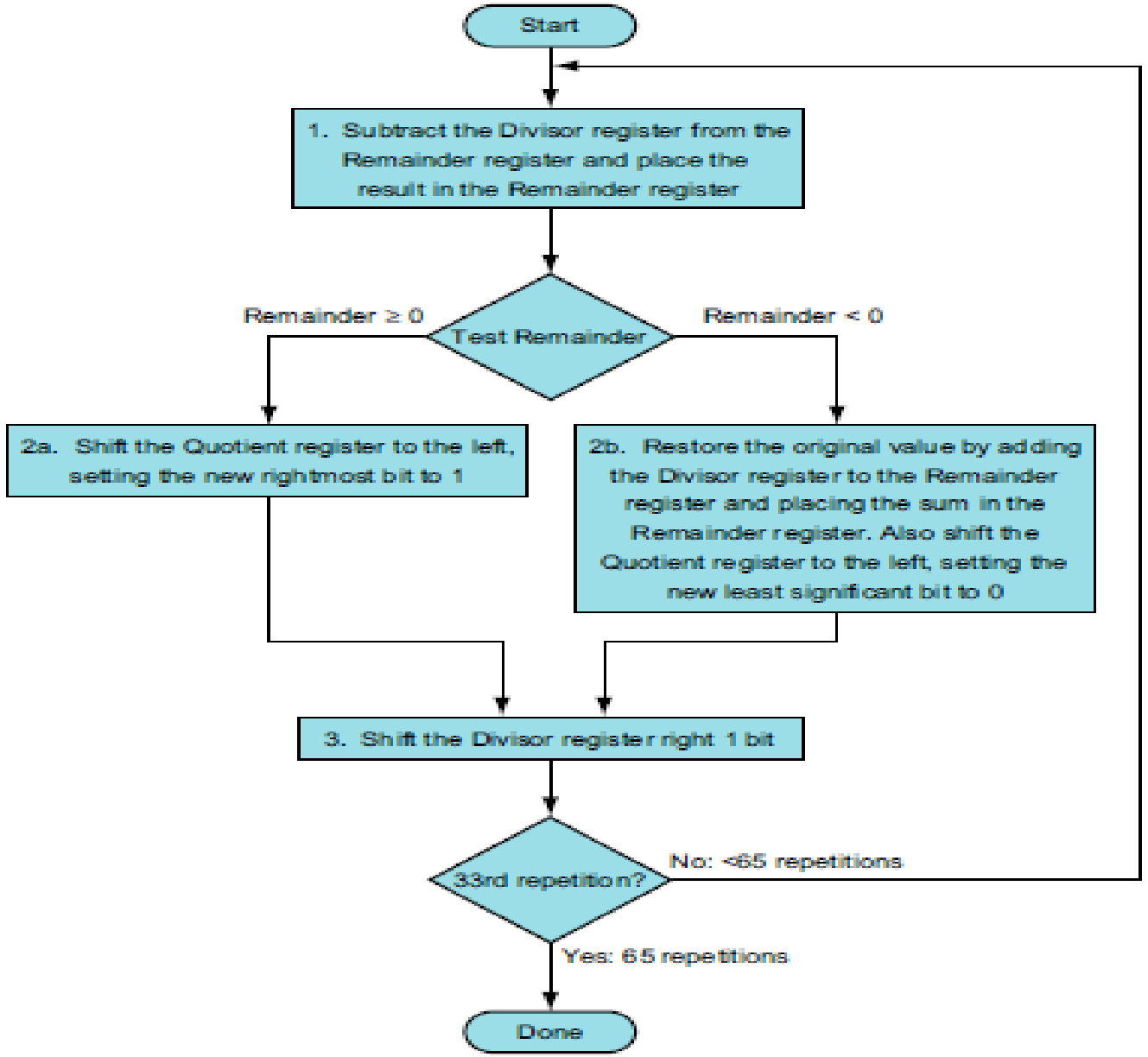
•The 32-bit divisor starts in the left half of the Divisor register and is shifted right 1 bit each iteration.

• The remainder is initialized with the dividend.

•Control decides when to shift the Divisor and Quotient registers and when to write the new value into the Remainder register.

Unit 4: Arithmetic for Computers

A Divide Algorithm:



A Divide Algorithm- Steps

Divide 7ten by 2ten: 0000 0111two by 0010 two.

•If the remainder is positive, the divisor did go into the dividend, so step 2a generates a 1 in the quotient.

• A negative remainder after step 1 means that the divisor did not go into the dividend, so step 2b generates a 0 in the quotient

and adds the divisor to the remainder, thereby reversing the subtraction of step 1.

•The final shift, in step 3, aligns the divisor properly, relative to the dividend for the next iteration. These steps are repeated 33 times.

Division example using the above algorithm

The bit to be examined to determine the next step is circled in color.

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: 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	1: 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	1: 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	1: Rem = Rem – Div	0000	0000 0100	⓪000 0011
	2a: Rem ≥ 0 ⟹ SLL Q, Q0 = 1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011
5	1: Rem = Rem – Div	0001	0000 0010	⓪000 0001
	2a: Rem ≥ 0 ⟹ SLL Q, Q0 = 1	0011	0000 0010	0000 0001
	3: Shift Div right	0011	0000 0001	0000 0001

THANK YOU

Vanamala H R

Department of Electronics and Communication

PES UNIVERSITY

CELEBRATING 50 YEARS