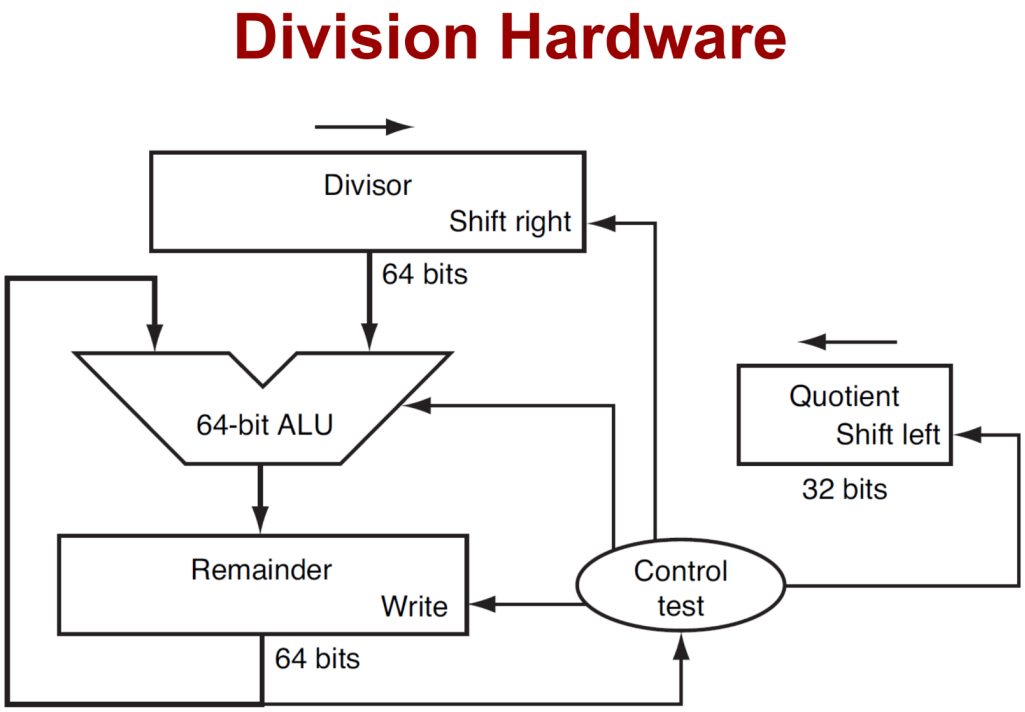
Binary Arithmetic Operations: Division and Multiplication

1 Binary Division Example: Restoring Division

Goal

Divide a binary number (dividend) by another (divisor) using the restoring division algorithm.

1.1 Sequential Division



Concept

- Initialize the remainder with the dividend.
- Subtract the divisor from the remainder.
- If the result is negative, restore the previous remainder and shift quotient bit as 0.

- If the result is non-negative, keep it and shift quotient bit as 1.
- Repeat with right-shifted divisor.

Example: $7 \div 2$

$$\text{Dividend } (7) = 0000\ 0111$$
$$\text{Divisor } (2) = 0010\ 0000 \quad \text{(initially left-shifted)}$$

Iteration	Quotient	Divisor	Remainder
0 (Init)	0000	0010 0000	0000 0111
1	0000	0001 0000	0000 0111 (Restored)
2	0000	0000 1000	0000 0111 (Restored)
3	0000	0000 0100	0000 0111 (Restored)
4	0001	0000 0010	0000 0011
5	0011	0000 0001	0000 0001

Result:

- Quotient = 0011 (3 in decimal)
- Remainder = 0000 0001 (1 in decimal)

$$\boxed{7 \div 2 = 3 \text{ remainder } 1}$$

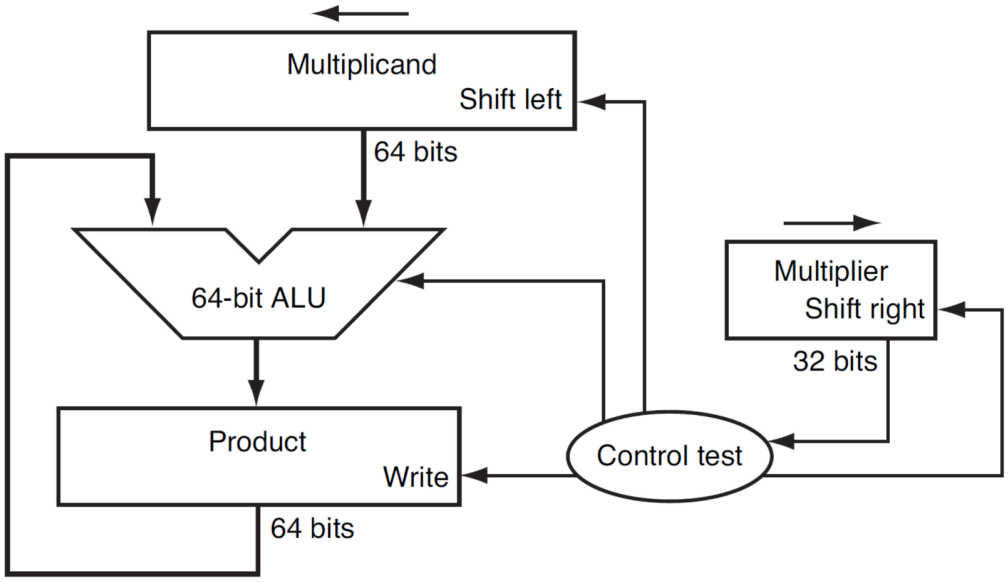
2 Binary Multiplication Example: Sequential Shift-and-Add

Goal

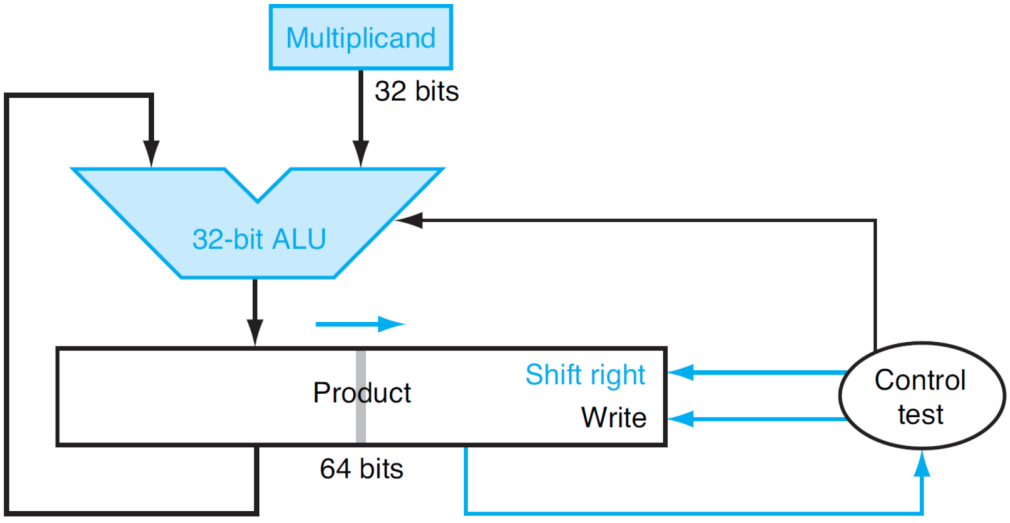
Multiply two binary numbers using shift-and-add sequential multiplication.

2.1 Sequential Multiplication

Sequential Multiplication Hardware



Refined Version of Multiplication Hardware



Concept

- If LSB of multiplier = 1, add multiplicand to product.
- Shift multiplicand left (multiply by 2).
- Shift multiplier right (divide by 2).
- Repeat for number of bits in multiplier.

Example: 2×3

$$\text{Multiplicand } (2) = 0000\ 0010$$
$$\text{Multiplier } (3) = 0011$$
$$\text{Product (Init)} = 0000\ 0000$$

Iteration	Multiplier	Multiplicand	Product
0 (Init)	0011	0000 0010	0000 0000
1	0001	0000 0100	0000 0010
2	0000	0000 1000	0000 0110
3	0000	0001 0000	0000 0110
4	0000	0010 0000	0000 0110

Result:

$$\boxed{2 \times 3 = 6}$$