4.4 COMMON SUBEXPRESSION ELIMINATION

A common subexpression is an expression that appears more than once and produces the same result.

Common Subexpression Elimination (CSE) is an optimization technique that removes repeated calculations by computing the expression only once and reusing its result.

Example:

Original Code (Before Optimization):

a = b + c

d = b + c

e = a * d

Here, the expression b + c appears twice. So, the same calculation is performed two times.

Optimized Code (After Optimization):

t = b + c

a = t

d = t

e = a * d

Now, b + c is calculated only once, and the result is reused using a temporary variable t.

How It Works

The compiler performs the following steps:

1. Finds expressions that occur repeatedly.
2. Checks whether the operands have changed.
3. If the operands have not changed, the previous result can be reused.
4. Creates a temporary variable if necessary.
5. Replaces the repeated expression with the temporary variable.

4.4.1 Important Condition

Common Subexpression Elimination is safe only when the operands have not changed between the two expressions.

Example:

a = b + c

b = 10

d = b + c

Here, b + c cannot be reused because the value of b has changed. Therefore, the compiler must calculate the expression again.

Benefits of Common Subexpression Elimination

1. Reduces repeated calculations.
2. Reduces program execution time.
3. Reduces the number of instructions.
4. Improves overall program performance.
5. Makes the code more efficient.