OPTIMIZATION PHASE

WHAT IS OPTIMIZATION?

Optimization can be assumed as something removes unnecessary code lines, and arranges the sequence of statements in order to speed up the program execution without wasting resources. In other words to make the program more efficient.

Optimization is performed by compiler in 2 ways:

- Machine dependent optimization
- Machine independent optimization

Machine dependent optimization

It is related to code generation and it is incorporated into code generation phase.

[Code generation phase: it produces object code in machine language]

Machine independent optimization

It can be performed independently of the target machine for which the compiler is generation a code that is the optimization are not tied to the target machines, specific platforms or languages.

DATA BASE

Matrix

- The major database used in optimization phase is matrix,which consists off operator,operands,forward and backward pointers.
- Forward pointers are the index of the next matrix entry in the program.
- Backward pointers are the index of the previous one.

operator	Operand 1	Operand 2	Forward ptr	Backward ptr

Identifier table

- This table is accessed to elet unwanted temporary storage and obtain information about identifiers.

Identifier name	Data attributes	Precedence

Literal table

- New literals are created by optimization phase and added into the literal table.

literal	base	scale	precision	Other information	address

ALGORITHM

There are 4 optimization techniques:

- Elimination of common sub-expression
- Compiler time computer
- Boolean expression optimization
- Move invariant computation outside of loops

Elimination of common sub-expression

- If there is identical sub-expression in the same statement.
- Elimination algorithm is used to delete one of them.

step 1: place the matrix in the form so that common sub-expression can be recognized

step 2: recognize 2 sub-expression are equal

step 3: eliminate one of them

step 4: alter the rest of the matrix to reflect the elimination of those entry

Compile time compute

- During computation involving constants at compile time a separate algorithm is used to perform that computation first and the execute the program.
- This will save both space and memory.

Example: A = 2 * 276 / 92 * B

A = 2 * 3 * B

A = 6 * B

Matrix before optimization

Matrix line	Operator	Operand 1	Operand 2
M1	/	276	92
M2	*	M1	2
M3	*	M2	B
M4	=	A	M3

Boolean expression optimization

- We use properties of boolean expression to shorten their computation.

Example:

- If a,b and c are boolean expression a statement if a OR b OR c if any one boolean expression is true, then the rest of the expression are not computed.

Move invariant computation outside of loops

- If a computation involves a variable that does not change within the loop it is called an invariant. [constant]
- All invariants are moved outside the loop before code generation.

```
Example: Do I=1  to 10
                out loop :
                        Do J=1 to 10;
                        LA  A=10
                              B=y(I+2);
                        LC C=C+10;
                         END
               END
```