



G H Patel College of Engineering & Technology

Gujarat Technological University

Submitted To: -

Prof. Namrata Dave

Submitted By: -

Gandhi Vivek

(140110107017)



LOOP OPTIMIZATION

Code Optimization: -

- The code optimization can be significantly done in **loops** of the program.
- Specially inner loop is a place where program spends large amount of time.
- Hence if number of instruction are less in inner loop then the running time of the program will get decreased to a large extent.
- Hence loop optimization is a technique in which code optimization performed on inner loops.

Methods of Loop Optimization: -

1. Code Motion
2. Induction Variable and Strength Reduction
3. Loop Invariant Method
4. Loop Unrolling
5. Loop Fusion

Code Motion: -

- Code motion is a technique which moves the code outside the loop.
- If there lies some expression in the loop whose result remain unchanged even after executing the loop several times, then such an expression should be placed just before the loop.

Code Motion: -

- For Example: -

```
while( i <= max - 1 )  
{  
    sum = sum + a[i];  
}
```

- Optimized Code: -

```
n = max - 1;  
while( i <= n )  
{  
    sum = sum + a[i];  
}
```

Induction Variable and Strength Reduction: -

- A variable x is called an induction variable of loop L if the value of variable gets changed every time.
- It is either decremented or incremented by some constant.

Induction Variable: -

- For Example: -

B₁

i := i + 1

t₁ := 4*j

t₂ := a[t₁]

If t₂ < 10 goto B₁

- In above code the value of i and t₁ are in locked state.
- That is, when value of i gets incremental by 1 then t₁ gets incremented by 4.
- Hence i and t₁ are induction variables.

Strength Reduction: -

- The strength of certain operators is higher than others.
- For Example: -
 - Strength of 8 is higher than +.

```
for ( i = 1; i <= 50; i++)  
    count = i * 7;
```

- Here we get values of count as 7, 14 and so on.

```
temp = 7;  
for ( i = 1; i <= 50; i++)  
{  
    count = temp;  
    temp = temp + 7;  
}
```

Loop Invariant Method: -

- In this optimization technique the computation inside the loop is avoided and there by the computation overhead on compiler is avoided.
- This ultimately optimizes code generation.

Loop Invariant Method: -

- For Example: -

```
for i := 0 to 10 do begin
```

```
  K = i + (a/b);
```

```
  ...
```

```
  ...
```

```
end;
```

- Can be written as

```
t := a/b;
```

```
for i := 0 to 10 do begin
```

```
  K = i + t;
```

```
  ...
```

```
  ...
```

```
end;
```

Loop Unrolling: -

- For Example: -

```
int i = 1;
While ( i <=100 )
{
    a[i] = b[i];
    i++;
}
```

- Can be written as

```
int i = 1;
While ( i <=100 )
{
    a[i] = b[i];
    i++;
    a[i] = b[i];
    i++;
}
```

Loop Fusion: -

- For Example: -

```
for i := 1 to n do
```

```
  for j := 1 to m do
```

```
    a[i, j] := 10
```

- Can be written as: -

```
for i := 1 to n*m do
```

```
  a[i] := 10
```

