OPTIMIZATION STRATEGIES

Odaloš Matej, Bc. xodalo00@stud.fit.vutbr.cz
Polesný Ondřej, Bc. xpoles01@stud.fit.vutbr.cz

December 13, 2012

Introduction

- Optimizer makes a more efficient version of the itermediate or target code
- **Local** optimization vs **global** optimization
- Optimization for **speed** vs optimization for **size**

Optimization categories

- Parser optimizations
- Linear peephole optimizations
- Structural optimizations

Parser optimizations

- Can be done by the parser itself
- Generate good code to begin with

Techniques

- Using logical lvalues rather than physical ones
- Minimizing the number of goto branches
- Instrinsic functions (i.e. math functions – sin, cos, sqrt)

Linear Peephole Optimizations

- Cannot be done by the parser itself
- Necessary to examine several blocks of code (peephole)
- Performed over small set of instructions (window)

Strength Reduction

- Replaces operation with a more efficient one
- Main objective is to save machine clock cycles
- $x * 8$ can be done with $x \ll 3$
- $x/8$ can be done with $x \gg 3$
- Multiplication by small numbers replaced by multiple additions
- Multiplication by larger numbers
- $t0 *= 9$ can be replaced by $t0 * 8 + t0 = t0 \ll 3 + t0$
- Modification of *jump* or *goto* instructions to match machine specific version that is more efficient

Constant folding and propagation

- Can be done by parser in a limited way
- $x + 2 * 3$ is treated like $x + 6$
- $a + 1 + 3$ parser ☹ independent optimizer ☺
- Multiplication by 1, addition and subtraction of zero and shift by zero eliminated
- $y = 5; x = y$ replaced with $y = 5; x = 5$
- Assignment of a constant is more efficient than memory to memory copy
- Optimizer keeps track of the contents of all variables that contain constants
- $t0 = 1; t0 \mathrel{+}= 5; t1 = to$ replaced with $t0 = 1; t1 = 6$

Dead Variables and Dead Code

- $t0 = 1; t0 \mathrel{+}= 5; t1 = to$ replaced with $t0 = 1; t1 = 6$
- After replacement, t0 is dead variable
- Variable considered dead from the last usage till its‘ reinitialization
- $x = 5; y = x; x \mathrel{+}= 1; x = z$
- Dead assignment, variable is never used or modified
- Elimination of code, that cannot be reached or does nothing useful
- $if\ (0)\ do\_something();$

Hardware problems

- $while(*port)$

```
{
    *port = 1; //pulse the low bit of the outer port
    *port = 0;
    delay( );
}
```

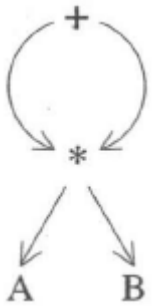
- Keyword volatile suppresses these optimizations

Structural optimizations

- Series of instructions ☹ parse or syntax tree ☺
- Parser generates intermediate language
- Intermediate code is processed by the optimizer

Common-Subexpression Elimination

- $A * B + A * B$ – subexpression eliminated twice
- Replaced by $t0 = A; t1 = B; t1 *= t0; t1 += t1$



Loop Unwinding

- Replaces the entire loop with the code that comprises the loop body, duplicated the number of times that the loop would execute