

**Метод инструментирования кода  
на этапе компиляции  
для направленной отладки  
оптимизирующих преобразований**

Д.А.Максименков  
АО “МЦСТ”

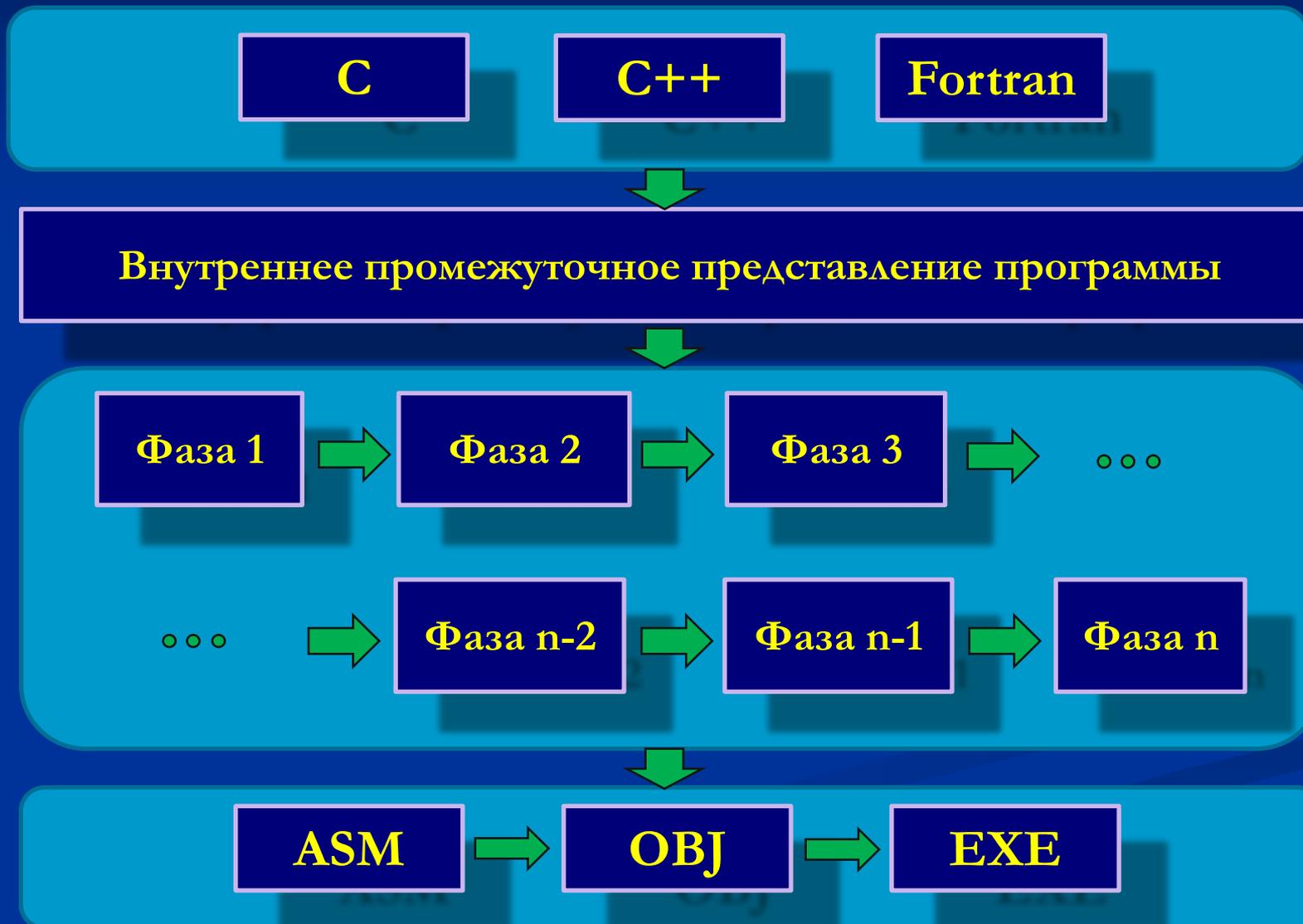
# Проблемы, возникающие при отладке оптимизирующих компиляторов

- ✓ Ограниченность тестового набора
- ✓ Ограниченность ресурсов тестирования
- ✓ Проблемы выявления факта ошибки оптимизирующего компилятора
- ✓ Локализация места ошибки при исполнении теста

# Методы инструментирования кода

- ✓ Готовые продукты для отладки:  
Valgrind, AddressSanitizer и др.
- ✓ Инструментирование исходного кода  
тестовых задач
- ✓ Встраивание динамических проверок в код  
задачи на этапе компиляции

# Механизм работы оптимизирующего компилятора



# Описание метода

а). Классическое преобразование `reerhole`:

```
b = a / 2;
```



```
b = a >> 2;
```

б). Преобразование `reerhole` с самопроверкой:

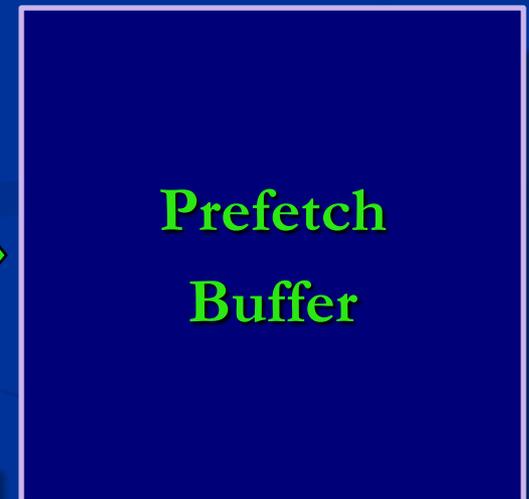
```
b = a / 2;
```



```
b = a >> 2;  
b1 = a/2;  
if (b != b1) Error(a);
```

# Оптимизация Array Prefetch Buffer (apb)

```
for ( i = 0; i < n; i++)  
    sum += a[ i ];
```



```
loop:  
    ld[ a + i ] → %reg  
    jmp loop
```



```
bap  
loop:  
    MOVA → %reg  
    jmp loop
```

# Применение метода при отладке оптимизирующего компилятора для процессоров “Эльбрус”

```
st [ mem1 ] ← %reg1  
...  
ld [ mem2 ] → %reg2
```



```
bap  
st [ mem1 ] ← %reg1  
...  
MOVA → %reg2
```

```
st [ mem1 ] ← %reg1  
...  
ld [ mem2 ] → %reg2
```



```
bap  
st [ mem1 ] ← %reg1  
...  
MOVA → %reg2  
ld [ mem2 ] → %reg3  
cmpe %reg2, %reg3 → %reg3  
ld [ %reg3 * mem2 ]
```

# Особенности предложенного метода

- ✓ Применим для широкого набора преобразований
- ✓ Позволяет производить направленную отладку тестируемой оптимизации
- ✓ Гарантированно проявляет ошибку при ее возникновении в оптимизированном коде
- ✓ Инструментирование кода задачи во время компиляции, а не на уровне исходных кодов:
  - не вносит возмущение в компилятор для применимости отлаживаемой оптимизации;
  - возможность использования для тестирования любых ВХОДНЫХ ДАННЫХ

# Результаты

- ✓ Предложенный метод был успешно реализован в оптимизирующих компиляторах для процессора “Эльбрус”
- ✓ За 8 месяцев тестирования выявлено более 20 ошибок надежности оптимизирующего компилятора
- ✓ Найдены 3 аппаратные ошибки в прототипах процессора “Эльбрус”