libcpu
Binary Translator

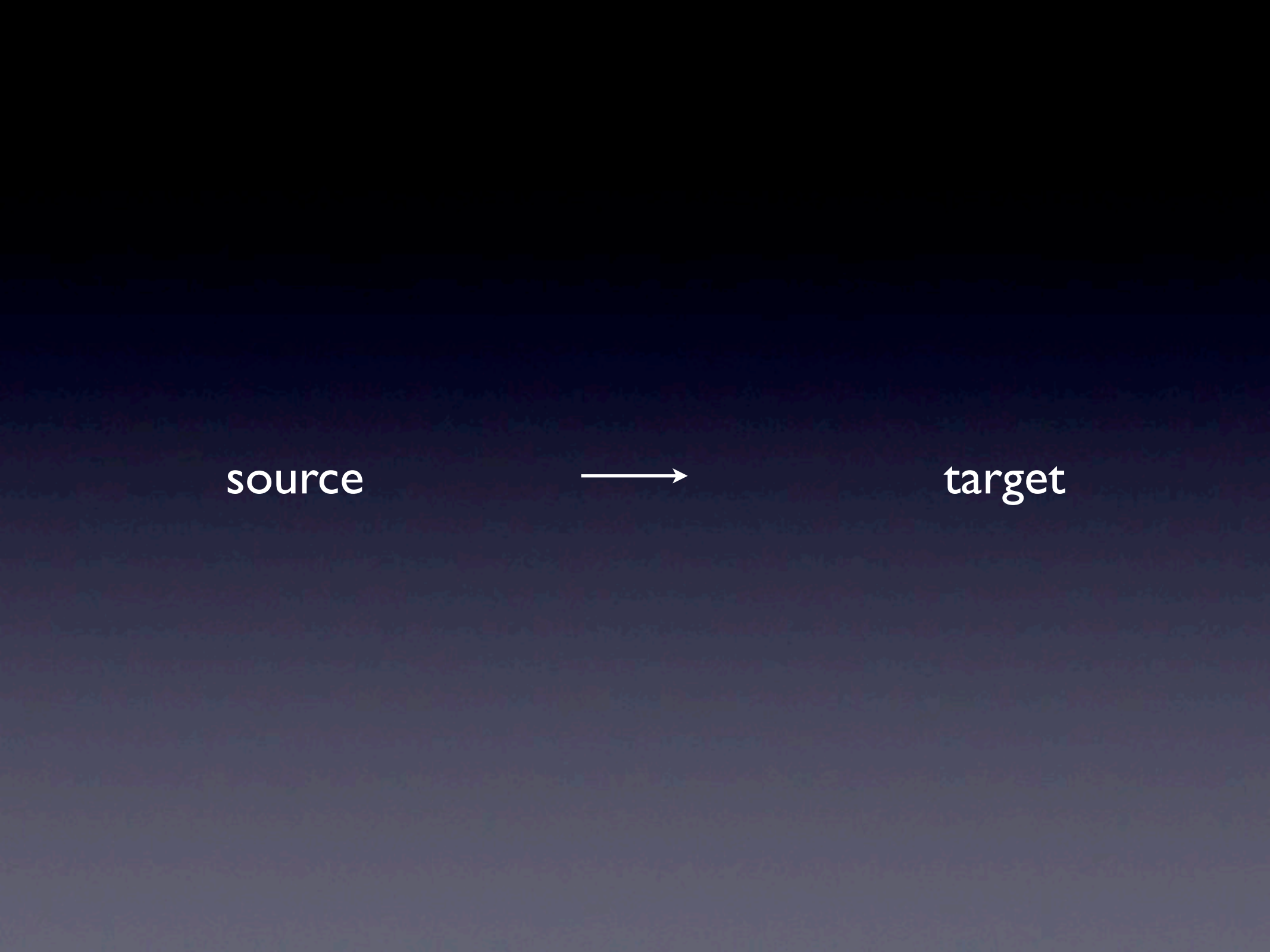
LLVM-based dynamic binary translation

Michael Steil
29 Dec 2009

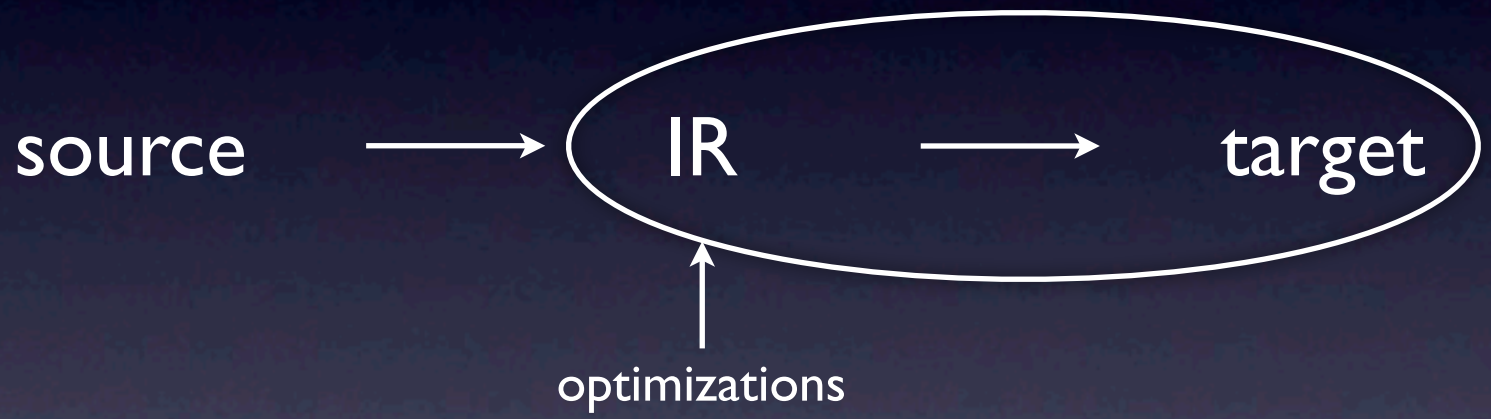
Chaos Communication Congress 26C3

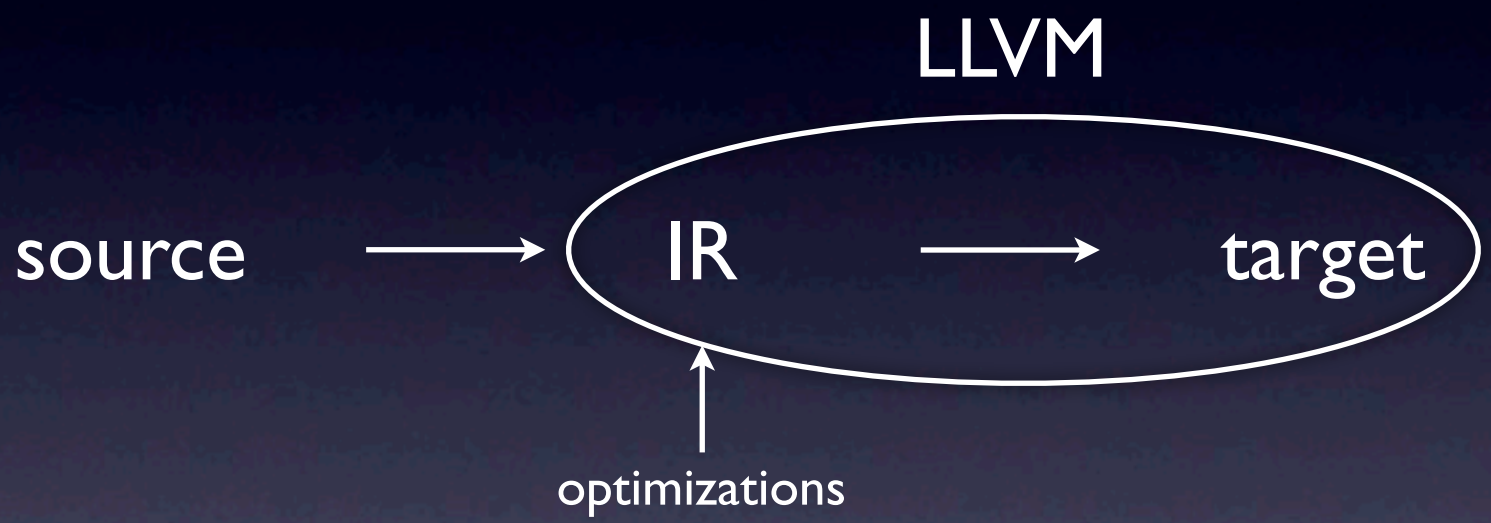
Binary Translation

- Translate machine code A into machine code B
- every modern emulator has one
  - PearPC: PowerPC to x86
  - UAE: M68K to x86
  - Nemu64: MIPS to x86


source
target


source
IR
target
optimizations


source
IR
target
optimizations


LLVM
source
IR
target
optimizations

LLVM

- compiler infrastructure
- C frontend (and others)
- static compiler & JIT
- intermediate representation on disk ("Bitcode")

libcpu

Generic **anything-to-anything** recompiler.

link against it in your emulator

```
cpu = cpu_new(CPU_ARCH_6502, 0, |
    CPU_6502_XXX_TRAP | CPU_6502_V_IGNORE);

cpu_set_flags_optimize(cpu, CPU_OPTIMIZE_ALL);
cpu_set_flags_debug(cpu, 0);
cpu_set_ram(cpu, RAM);

int ret = cpu_run(cpu, entry);
```

Core

Code Flow Analysis

Conditional Logic

LLVM Interface

Basic Block Caching

library interface

reusable!

Frontends

Motorola 88K

MOS 6502

ARM

MIPS

M68K

SPARC

just add more

Status

- M88K: *complete*

  OpenBSD user mode (think: perl, ftp, grep)
- 6502: *complete*

  Commodore 64 BASIC
- ARM, MIPS: small tests

Architecture Description Language

```
arch "8086" {

    name "Intel 8086";

    endian little;

    byte_size 8;
    word_size 16;
    address_size 24;
    psr_size 16;

    register_file {

        group R {

            [ #i16 bx -> #i8 ( bh : bl ) ],
            [ #i16 ax -> #i8 ( ah : al ) ],
            [ #i16 cx -> #i8 ( ch : cl ) ],
            [ #i16 dx -> #i8 ( dh : dl ) ],
            [ #i16 sp ],
            [ #i16 bp ],
            [ #i16 si ],
            [ #i16 di ]
        }
```

```
insn or   : dst = %CC ( dst | src );
insn xor  : dst = %CC ( dst ^ src );
insn neg  : dst = %CC ( - dst );
insn not  : dst = %CC ( ~ dst );
insn shl  : dst = %CC ( dst << src );
insn shr  : dst = %CC ( %U ( dst >> src ) );
insn sar  : dst = %CC ( %S ( dst >> src ) );
insn rol  : dst = %CC ( dst <<> src );
insn ror  : dst = %CC ( dst >>< src );
insn rcl  : dst = %CC ( dst ^<<> src );
insn rcr  : dst = %CC ( dst ^>>< src );
```

www.libcpu.org/wiki

Michael Steil <mist@c64.org>