LLVM – An Introduction

Linux Collaboration Summit, April 7, 2011

David Kipping, Qualcomm Incorporated

LLVM – An Introduction

LLVM Vision and Approach

- Primary mission: build a set of modular compiler components:
  - Reduces the time & cost to construct a particular compiler
    - A new compiler = glue code plus any components not yet available
  - Components are shared across different compilers
    - Improvements made for one compiler benefits the others
  - Allows choice of the right component for the job
    - Don’t force “one true register allocator”, scheduler, or optimization order
- Secondary mission: Build compilers that use these components
  - ... for example, an amazing C compiler

LLVM Umbrella of Projects

- LLVM Language independent optimizer and code generator
  - Many optimizations, many targets
  - Not a compiler by itself
- Clang C/C++/Objective-C front-end
  - Designed for speed, reusability, compatibility with GCC
- MC: Machine Code slicing and dicing
  - Assemblers, disassemblers, object file processing
- LLDB – LLVM Debuggers
  - Native debugger that reuses Clang's parser, LLVM JIT, MC disassemblers
- libc++: C++ standard runtime library
  - Full support for C++'0x

LLVM Code Generator Highlights

Approachable C++ code base, modern design, easy to learn

- Strong and friendly community, good documentation

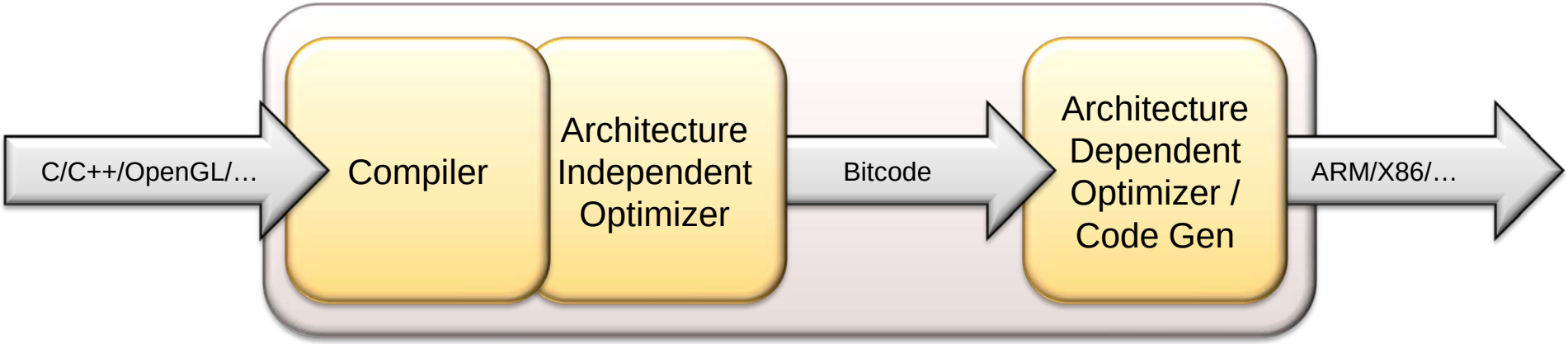
Language and target independent code representation

- Very easy to generate from existing language front-ends
- Text form allows you to write your front-end in perl if you desire

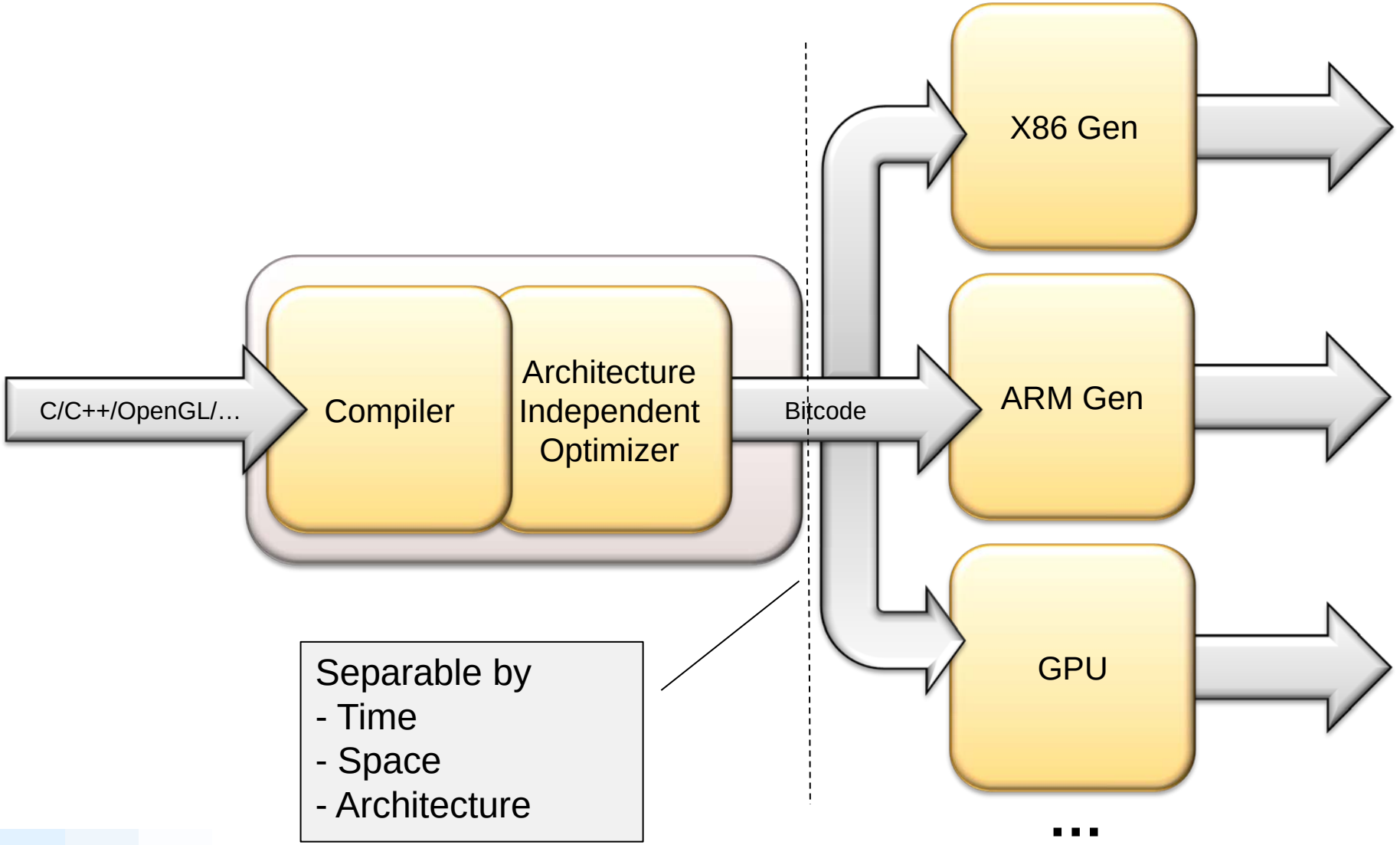
Modern code generator

- Easily retargetable to new chips
- Many popular targets supported:
  - X86, ARM, PowerPC, SPARC, Alpha, MIPS, Blackfin, CellSPU, MBlaze, MSP430, XCore, etc.
- Supports both JIT and static code generation

LLVM – Static Compiler Configuration



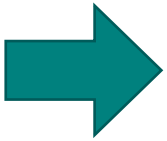
LLVM – JIT Configuration



Colorspace Conversion

- Code to convert from one color format to another:
  - e.g. BGRA 444R to RGBA 8888
  - Hundreds of combinations, importance depends on input

```
for each pixel {
  switch (infmt) {
  case RGBA5551:
    R = (*in >> 11) & C
    G = (*in >> 6) & C
    B = (*in >> 1) & C
  ... }
  switch (outfmt) {
  case RGB888:
    *outptr = R << 16 |
              G << 8 ...
  }
}
```



Run-time specialize

```
for each pixel {
  R = (*in >> 11) & C;
  G = (*in >> 6) & C;
  B = (*in >> 1) & C;
  *outptr = R << 16 |
            G << 8 ...
}
```

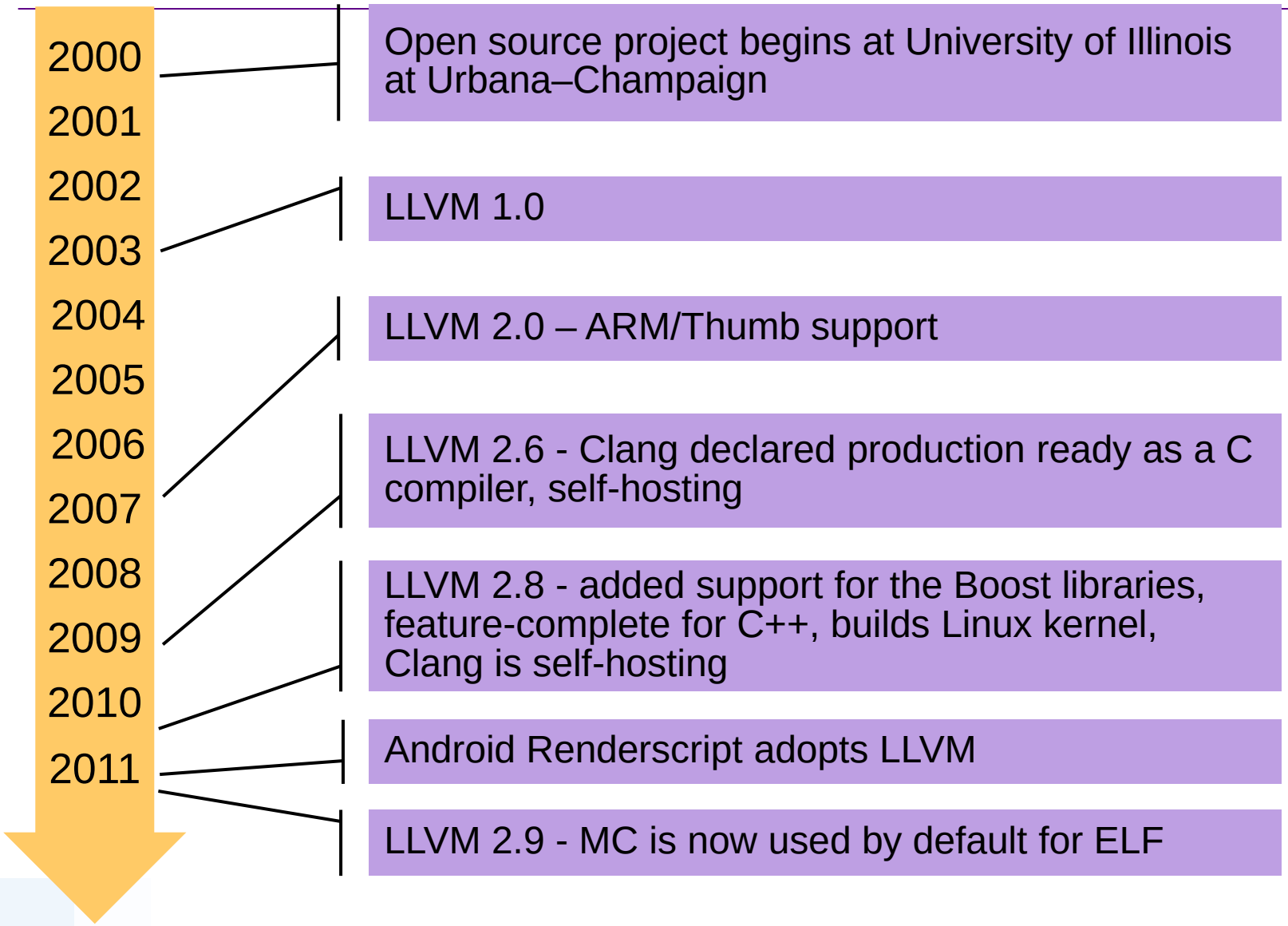
Compiler optimizes shifts and masking

- Speedup depends on src/dest format:
  - 5.4x speedup on average, 19.3x max speedup: (13.3MB/s to 257.7MB/s)

LLVM Growing Impact

- Solid technology base for tool evolution and code generation for the next decade
- Provides an excellent common ground for collaboration
  - Open source
  - Vibrant ecosystem
  - Release cadence of approximately 6 months
- LLVM has been adopted in important technologies and will become a standard component for Linux platforms
  - Chrome PNaCl
  - Android Renderscript
  - OpenCL – discussions on adopting LLVM IR
  - Graphics pipelines – prevalent
- LLVM will start shipping in volume on Linux mobile devices this year with Android Honeycomb

LLVM Timeline



Further Adoption

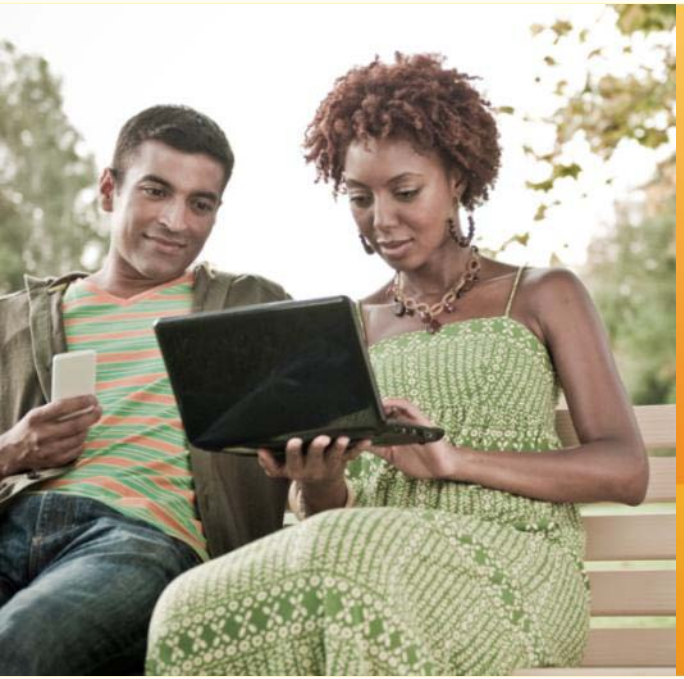
- Active investigations in other system areas adopting LLVM
- Potential to use LLVM for complete system builds
  - Reduce build complexities
  - Leverage bug fixes and optimizations
- LLVM is not quite ready for production Linux builds
  - Continued improvements in generated code performance
  - Further integration with Linux ecosystem
- If interested, join the LLVM community
  - LLVM.org
  - Next developers meeting in the fall

Agenda

Session	Speaker	Start	Duration
Introduction to LLVm	David Kipping	9:00	0:20
LLVM Use Cases - PNaCl	David Sehr	9:20	0:30
LLVM Use Cases - LLVM for RenderScript and Pixelflinger	Shih-wei Liao	9:50	0:30
Benchmarking & Continuous Testing of LLVM	Michael Larabel	10:20	0:30
Break		10:50	0:15
Building Linux with LLVM	Bryce Lelbach	11:05	0:30
LLVM Ecosystem	Mark Mitchell	11:35	0:35

Disclaimer

Nothing in these materials is an offer to sell any of the components or devices referenced herein. Certain components for use in the U.S. are available only through licensed suppliers. Some components are not available for use in the U.S.



Follow us on:

Thank You

For more information on Qualcomm, visit us at:
www.qualcomm.com
www.qualcomm.com/blog