

# Basics of GNU Compiler Collection (GCC)

- To: AMI India Engineers
- By: Chandra Kumar R
- Date: 14-May-2010

# Time Duration

- Estimated Time : ~45 minutes

# Sequence of Presentation

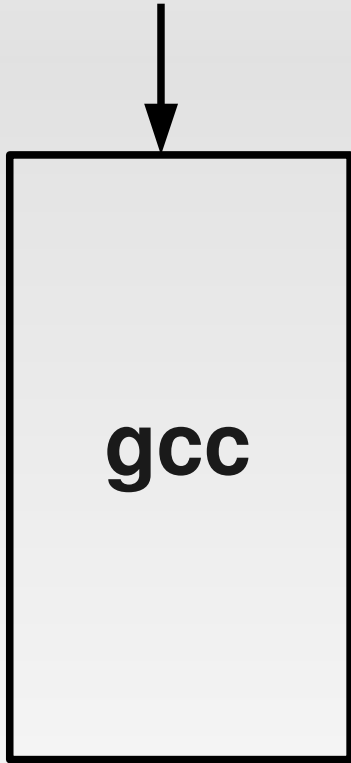
- Introduction on GCC
- The impact of retargetability
- Architecture of GCC
- Important phases of GCC
- Building GCC

# Introduction on GCC

- GCC stands for GNU Compiler Collection
- Compiler Generation Framework
- Resourceable and retargetable

# GNU Tool Chain

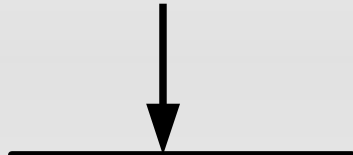
Source Program



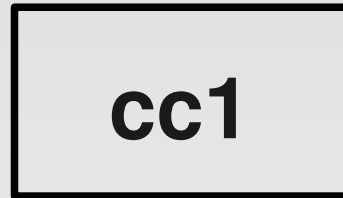
Target Program

# GNU Tool Chain

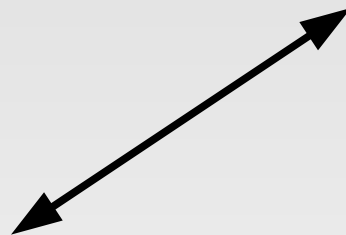
Source Program



**gcc**



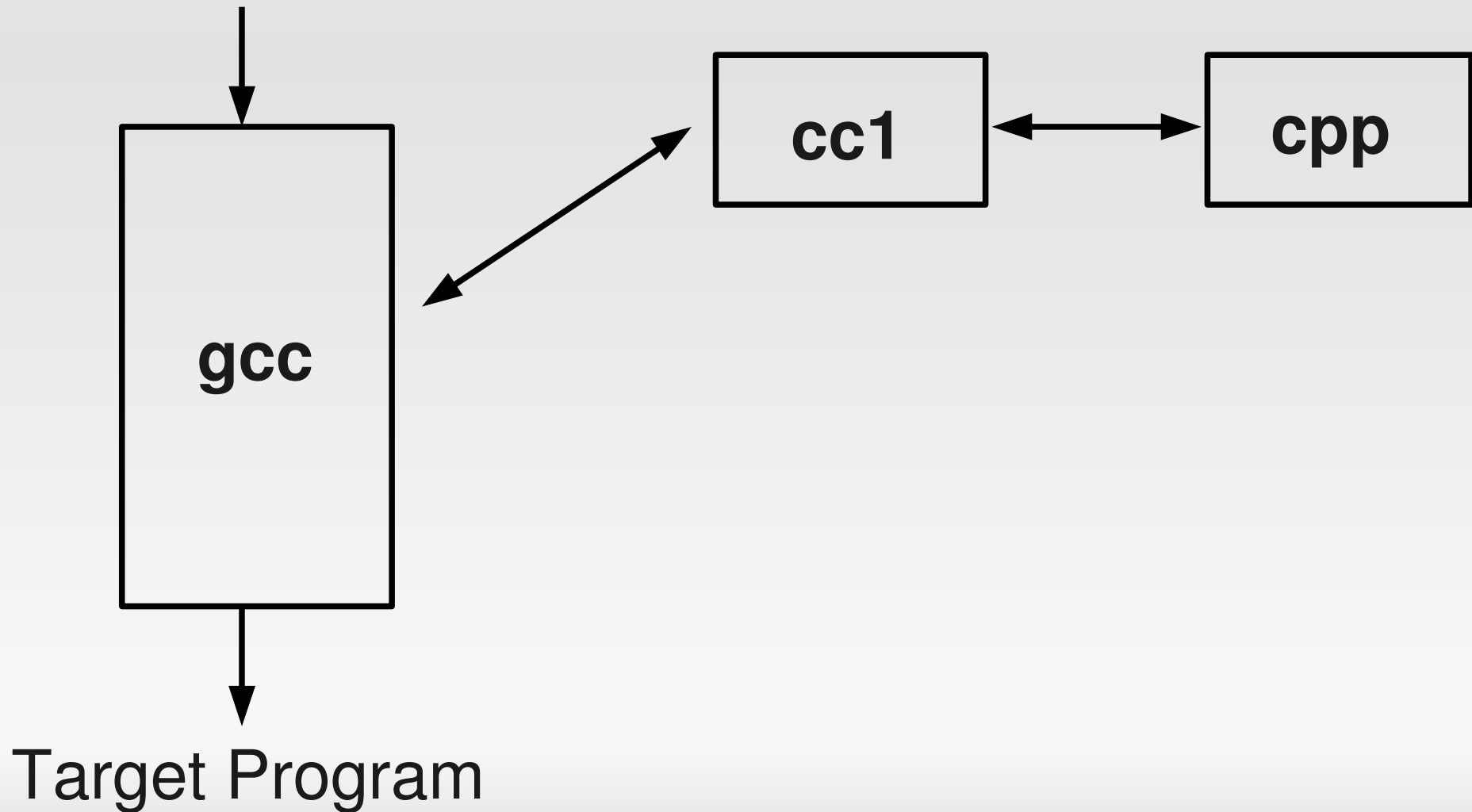
**cc1**



Target Program

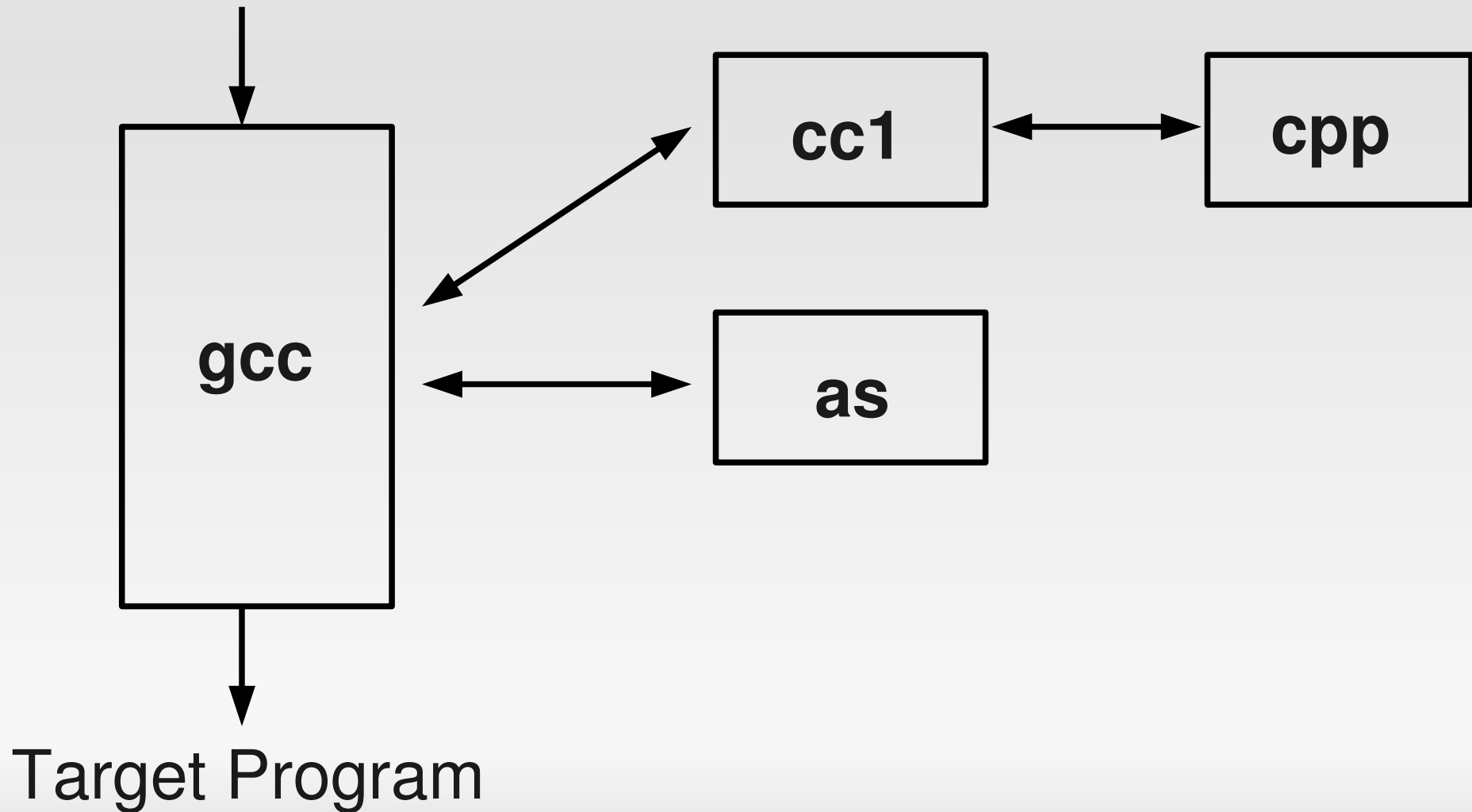
# GNU Tool Chain

Source Program



# GNU Tool Chain

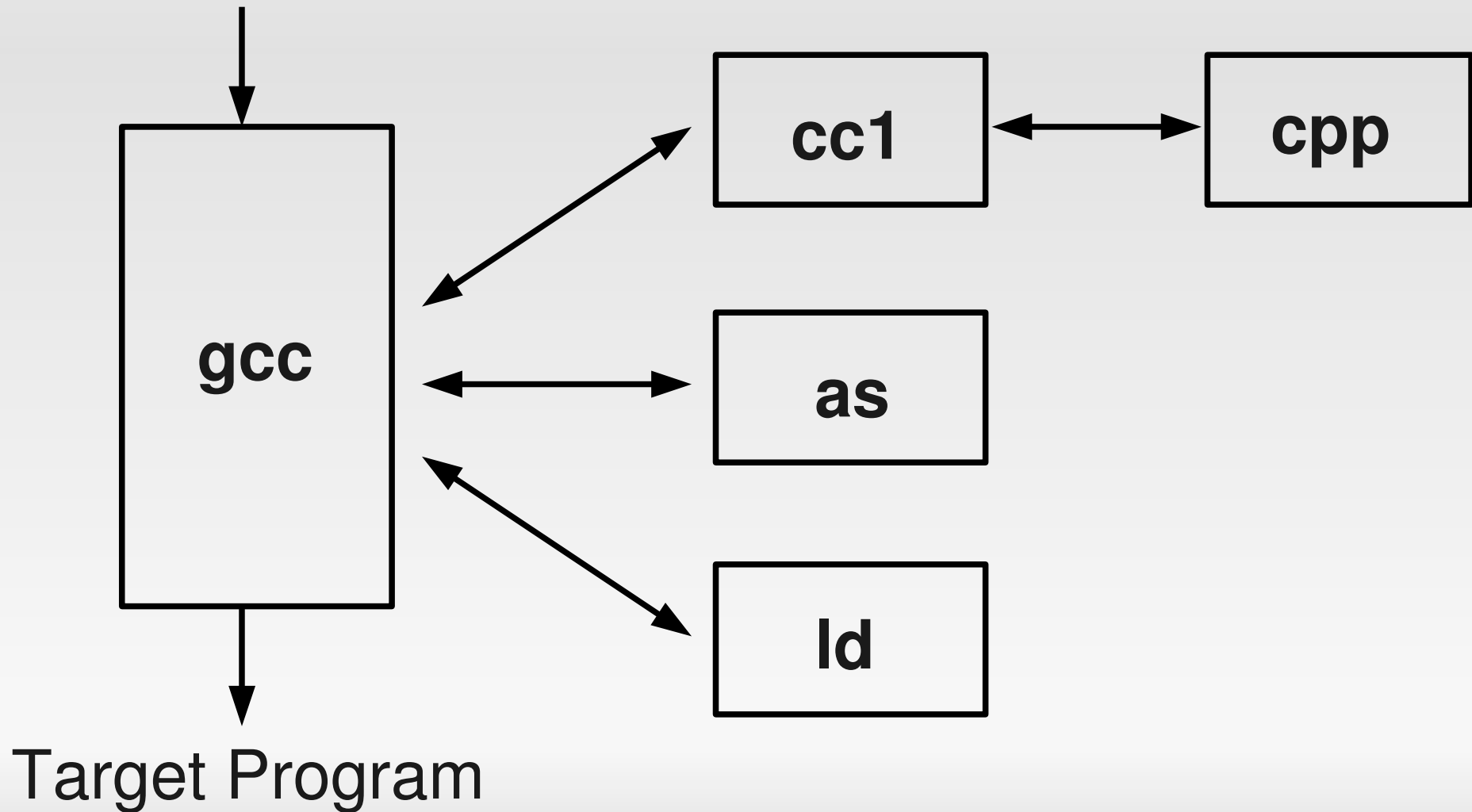
Source Program





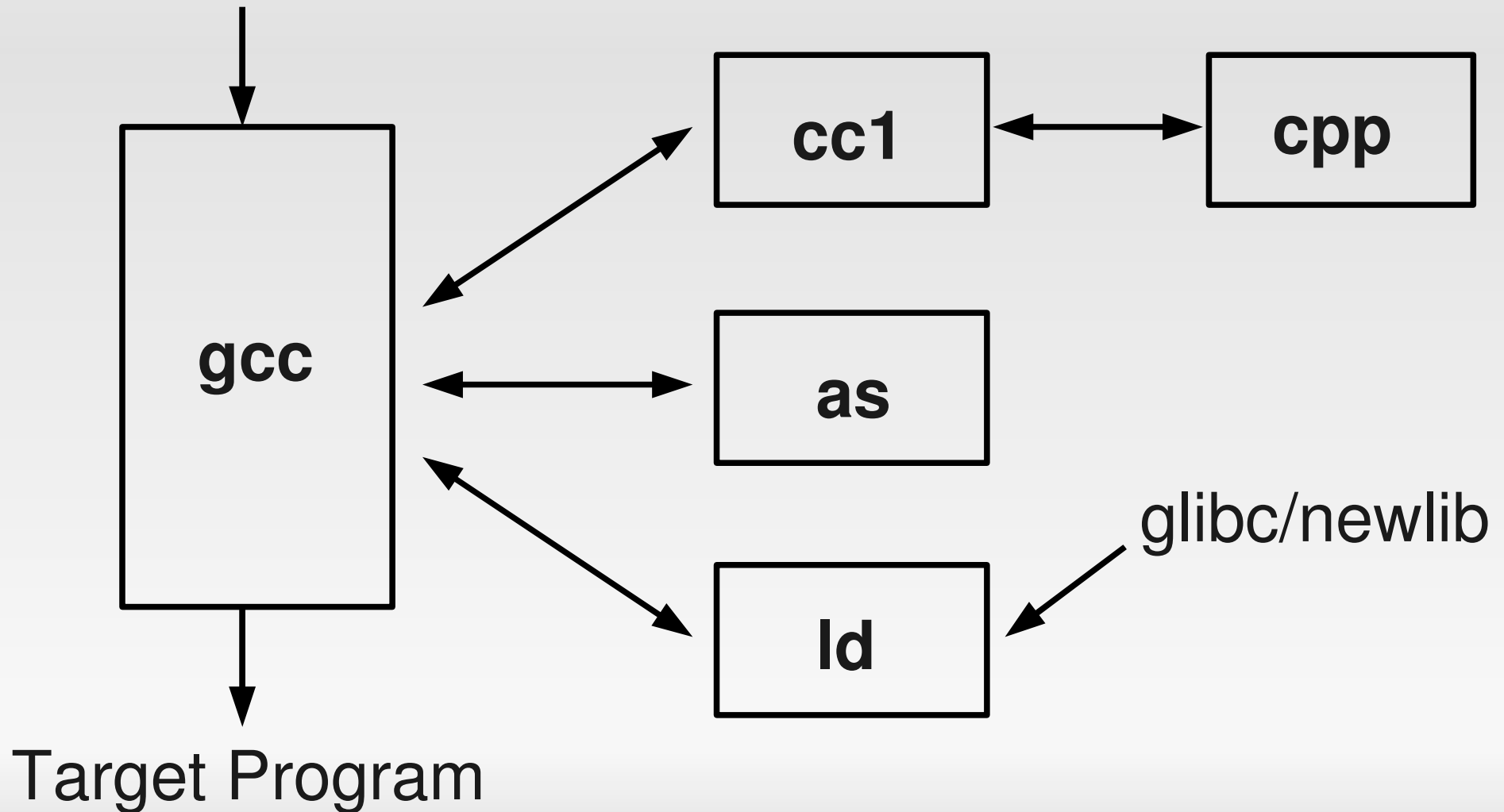
# GNU Tool Chain

Source Program



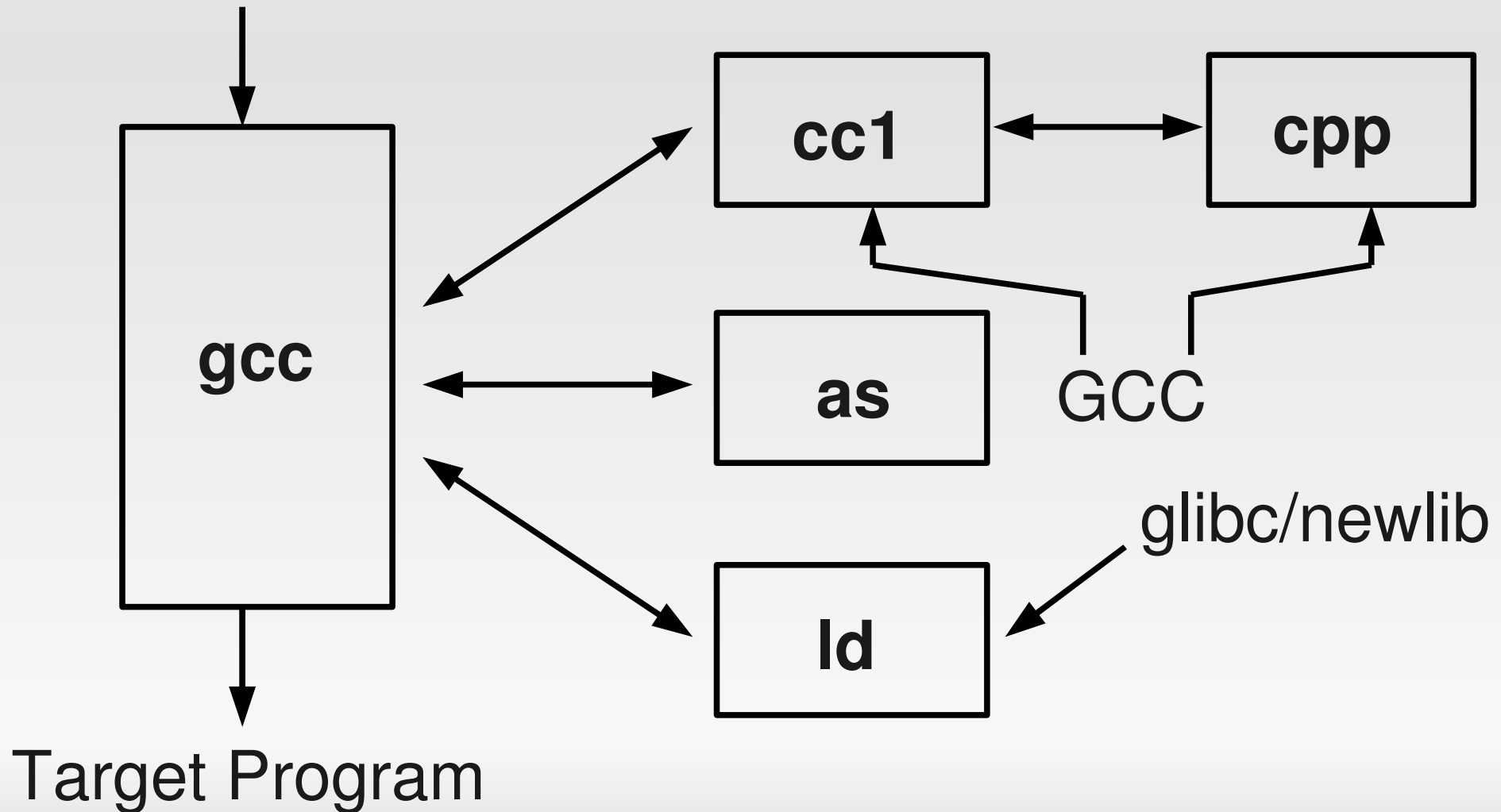
# GNU Tool Chain

Source Program



# GNU Tool Chain

Source Program

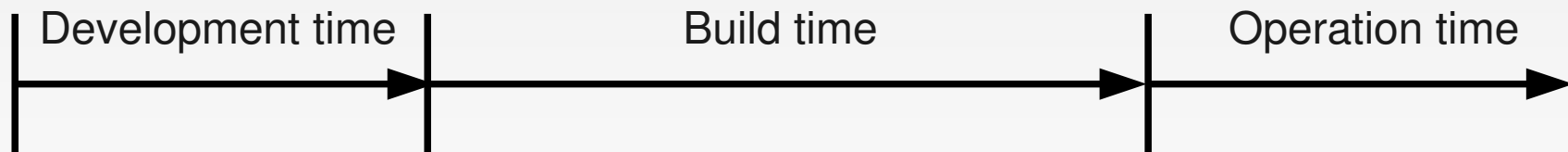


# Impact of retargetability

- Retargetability
  - Target specific information is not available at implementation time
  - Target specific information should be incorporated at build

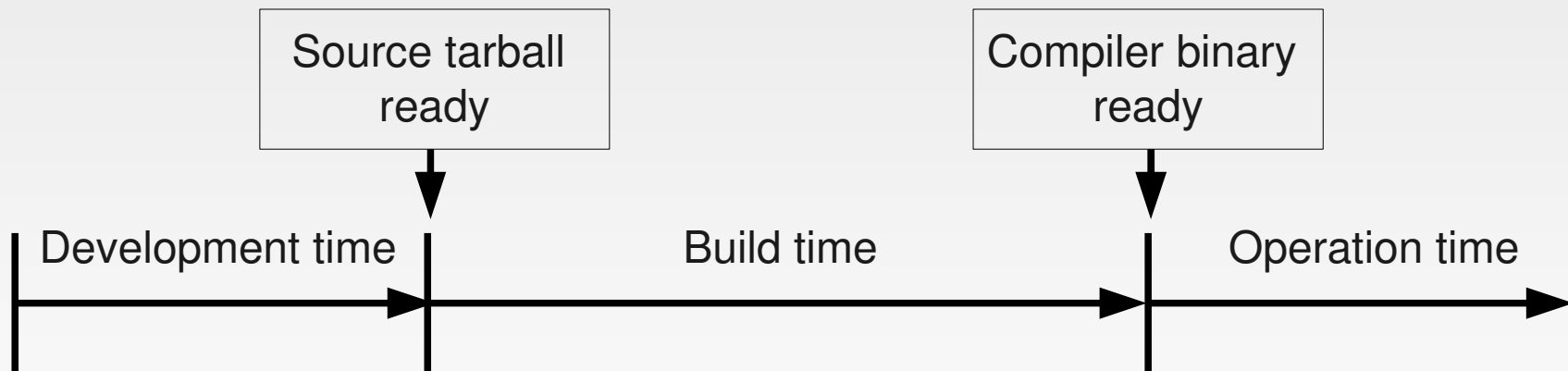
# Impact of retargetability

- Retargetability
  - Target specific information is not available at implementation time
  - Target specific information should be incorporated at build



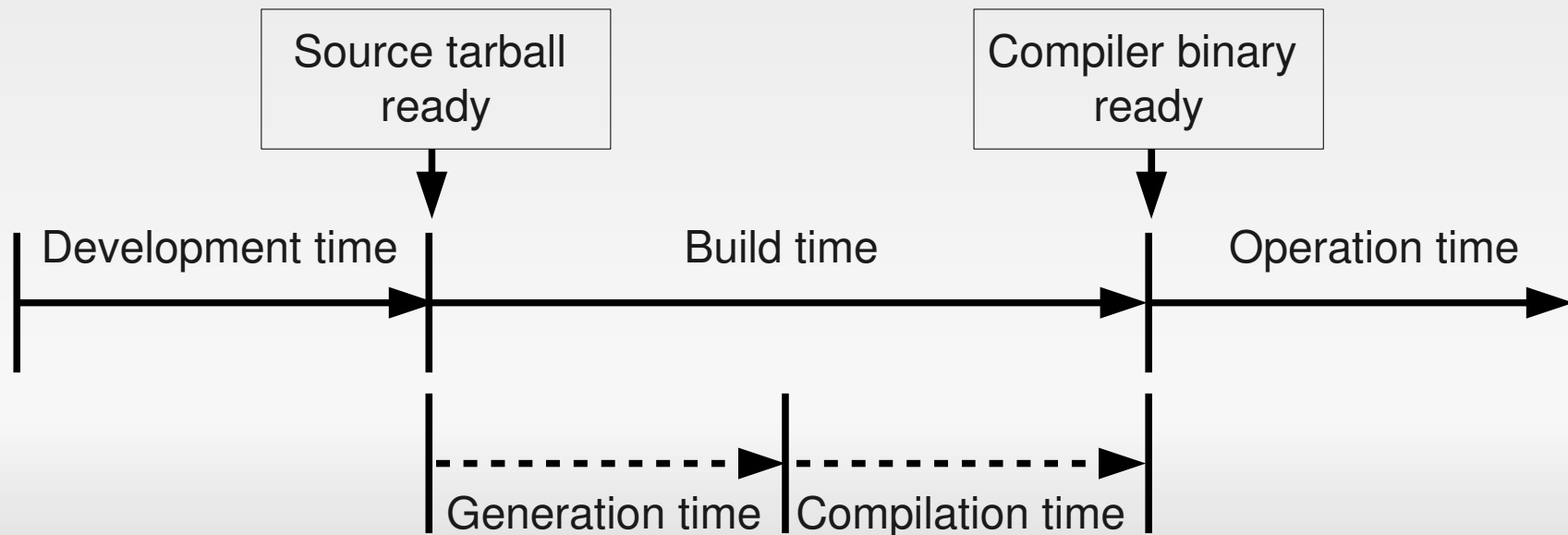
# Impact of retargetability

- Retargetability
  - Target specific information is not available at implementation time
  - Target specific information should be incorporated at build



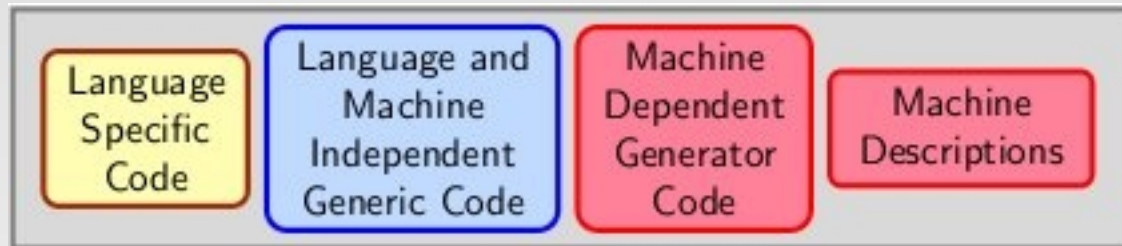
# Impact of retargetability

- Retargetability
  - Target specific information is not available at implementation time
  - Target specific information should be incorporated at build



# Architecture of GCC

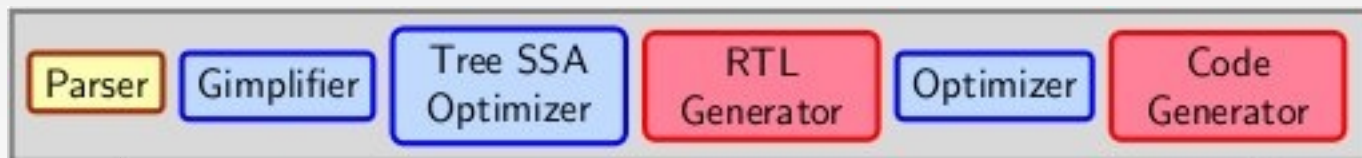
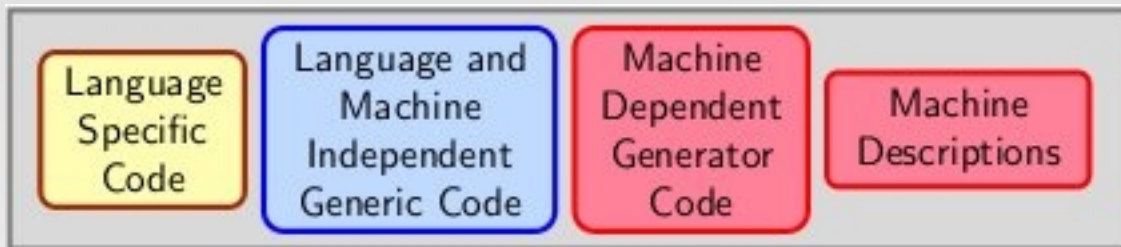
## Compiler Generation Framework





# Architecture of GCC

## Compiler Generation Framework



Source Program

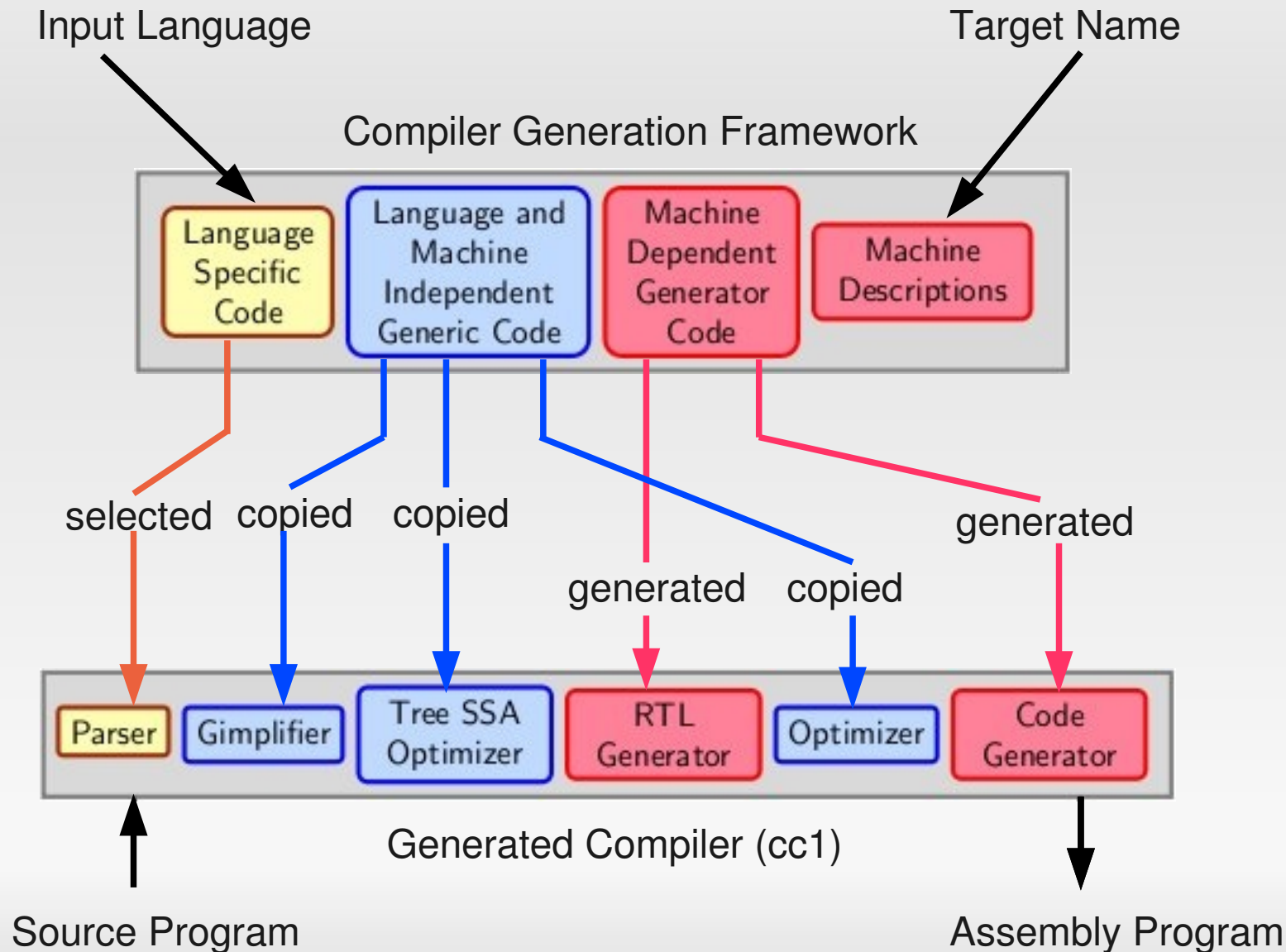
↑

Generated Compiler (cc1)

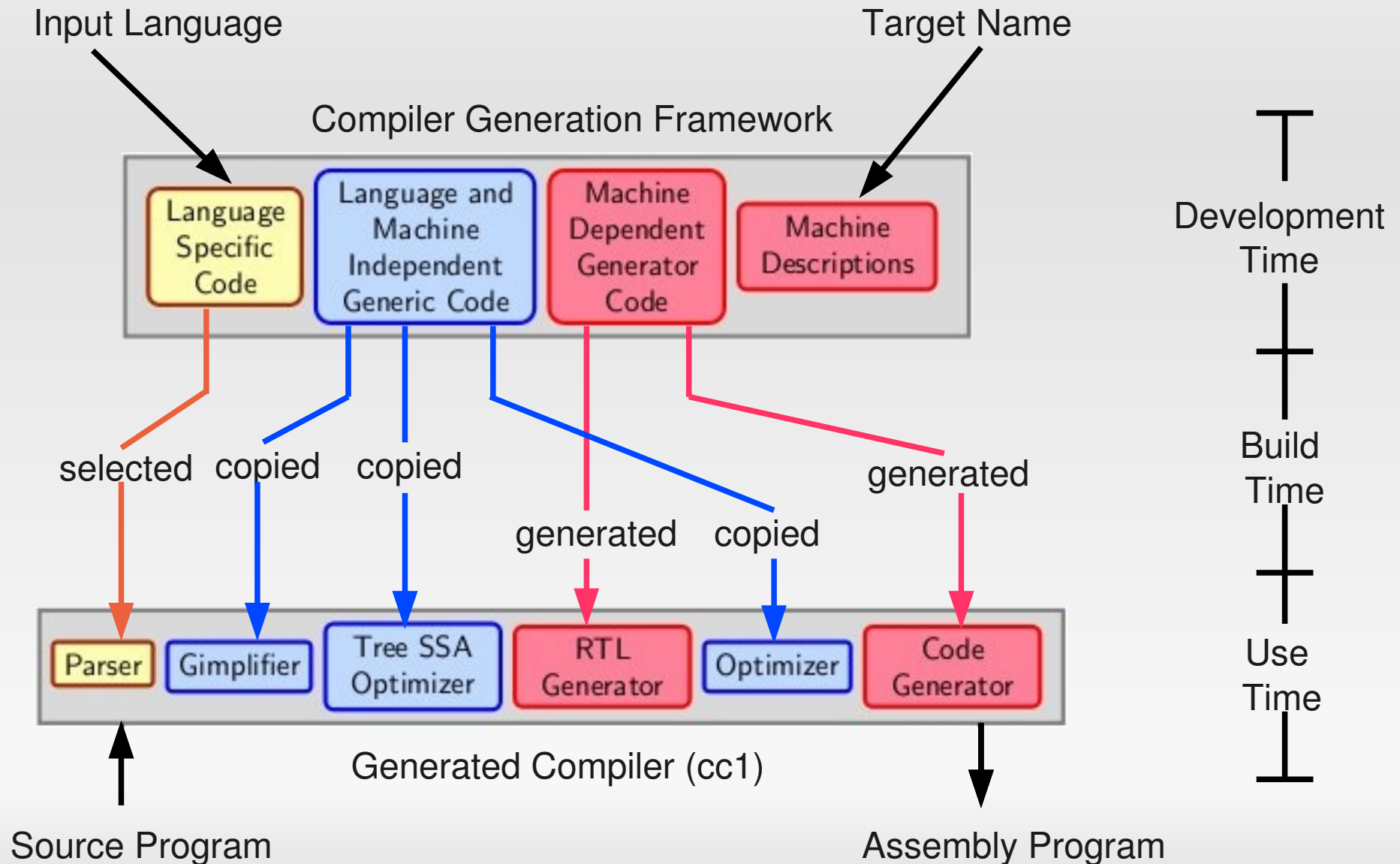
↓

Assembly Program

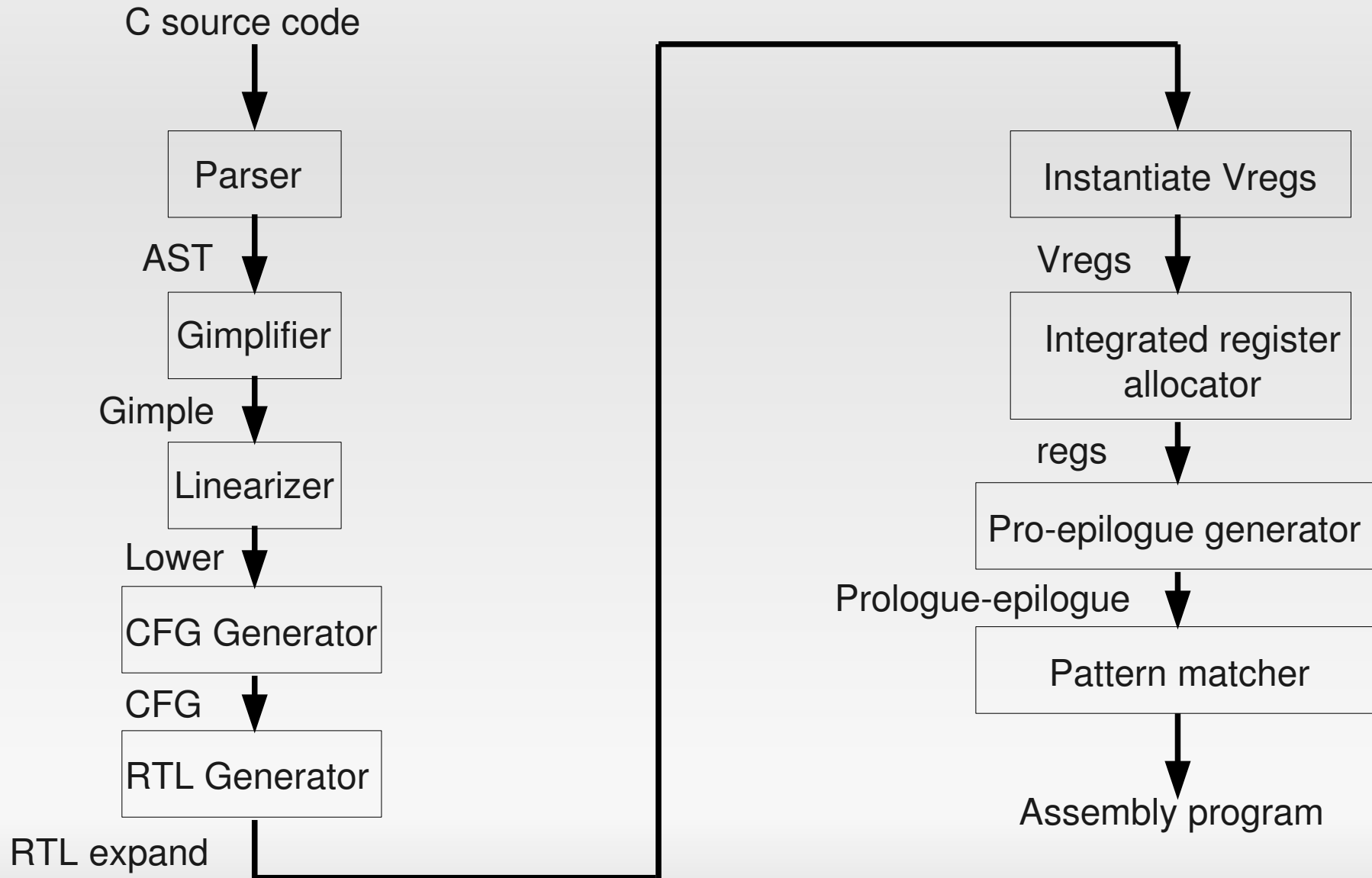
# Architecture of GCC



# Architecture of GCC



# Important Phases of GCC



# Important Phases of GCC

(continued...)

- -fdump-tree-<pass>
  - -fdump-tree-original
  - -fdump-tree-gimple
  - -fdump-tree-lower
  - -fdump-tree-cfg
- -fdump-rtl-<pass>
  - -fdump-rtl-expand
  - -fdump-rtl-vreg
  - -fdump-rtl-ira
  - -fdump-rtl-pro\_and\_epilogue

# Building GCC

- Variants of Compiler Builds
- Configuration
- Steps in Configuration and Building

# Variants of Compiler Builds

- Native Build
  - Build System = Host System = Target System
- Cross Build
  - Build System = Host System  $\neq$  Target System
- Canadian Cross
  - Build System  $\neq$  Host System  $\neq$  Target System

# Configuration

Preparing the GCC source for local adaptation:

- The platform on which it will be compiled
- The platform on which the generated compiler will execute
- The platform for which the generated compiler will generate code
- The directory in which the source exists



# Configuration (continued...)

- The directory in which the compiler will be generated
- The directory in which the generated compiler will be installed
- The input languages which will be supported
- The libraries that are required
- etc.

# Common Configuration Options

- `--target`
  - To specify the target during cross build
- `--enable-languages`
  - Comma separated list of language names
- `--prefix=<GCC install directory>`
  - toplevel installation directory. Toplevel installation directory defaults to `/usr/local`
- `--program-prefix`
  - Prefix string for executable names

# Steps in Configuration and Building

- Download and untar the source
- `cd $(BUILD)`
- `$(SOURCE)/configure`
- `make`
- `make install`

`$(SOURCE)` : GCC source directory

`$(BUILD)` : GCC build directory

**Thank You**