Workshop on Essential Abstractions in GCC

The Retargetability Model of GCC

GCC Resource Center
(www.cse.iitb.ac.in/grc)

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay

July 2009

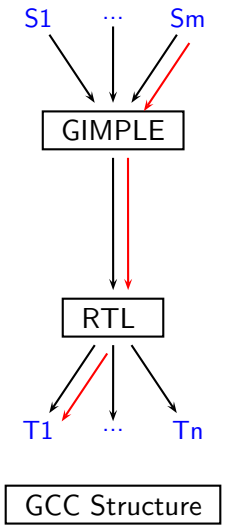
Outline

- A Recap
- Generating the code generators
- Using the generator code generators

Part 1

A Recap

Recapitulate: The GCC Build



Front end: Multiple source languages

- Separate HLL dependent part of code
- Selection mechanism required
- Parsers for each source
- Reduce to a common IR – GIMPLE

July 09 Retargetability Model: A Recap 2/15

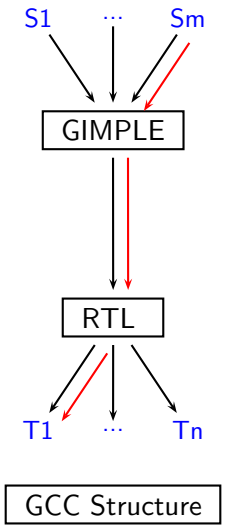
Recapitulate: The GCC Build

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

July 09 Retargetability Model: A Recap 2/15

Recapitulate: The GCC Build



Middle: Optimisations, translations

- Decide: placement in phase sequence
- Try: match optimiser needs & IR properties

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

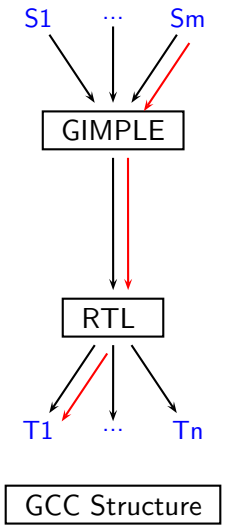
July 09 Retargetability Model: A Recap 2/15

Recapitulate: The GCC Build

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Recapitulate: The GCC Build



Back end: Multiple targets

- Separate target dependent part
- Description system for target props
- Linear IR preferable

Notes

July 09 Retargetability Model: A Recap 2/15

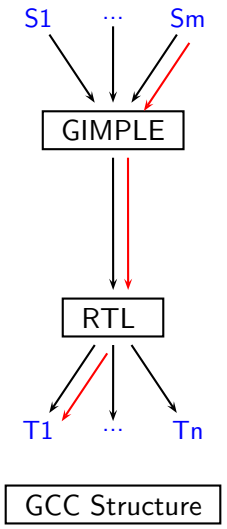
Recapitulate: The GCC Build

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

July 09 Retargetability Model: A Recap 2/15

Recapitulate: The GCC Build



GCC → gcc/cc1: Build:

- Select Input Language and Target Processor
- Generate target specific code+data
- Compile the generated code along with the common code

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

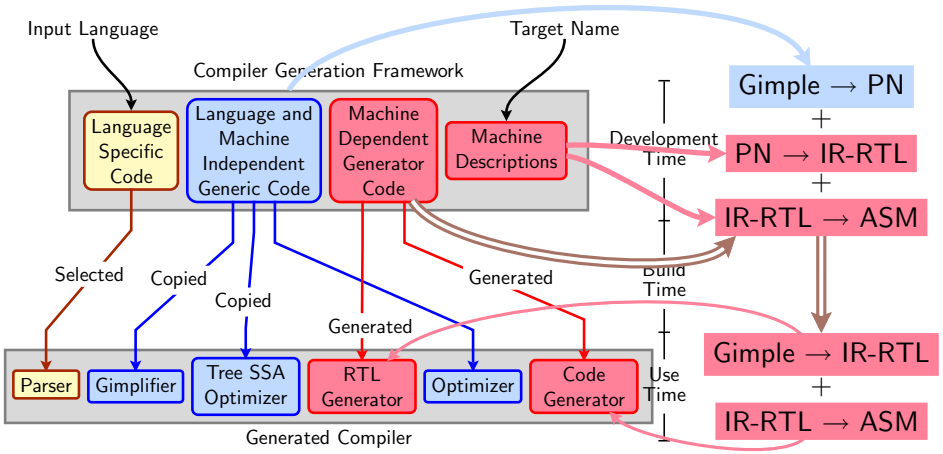
July 09 Retargetability Model: A Recap 2/15

Recapitulate: The GCC Build

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Retargetability Mechanism of GCC



Retargetability Mechanism of GCC

Notes

What is "Generated"?

- Info about instructions supported by chosen target, e.g.
  - Listing data structures (e.g. instruction pattern lists)
  - Indexing data structures, since diff. targets give diff. lists.
- C functions that generate RTL internal representation
- Any useful "attributes", e.g.
  - Semantic groupings: arithmetic, logical, I/O etc.
  - Processor unit usage groups for pipeline utilisation

What is "Generated"?

Notes

Information supplied by the MD

- The target instructions – as ASM strings
- A description of the semantics of each
- A description of the features of each like
  - Data size limits
  - One of the operands must be a register
  - Implicit operands
  - Register restrictions

Information supplied	in `define_insn` as
The target instruction	ASM string
A description of it's semantics	RTL Template
Operand data size limits	predicates
Register restrictions	constraints

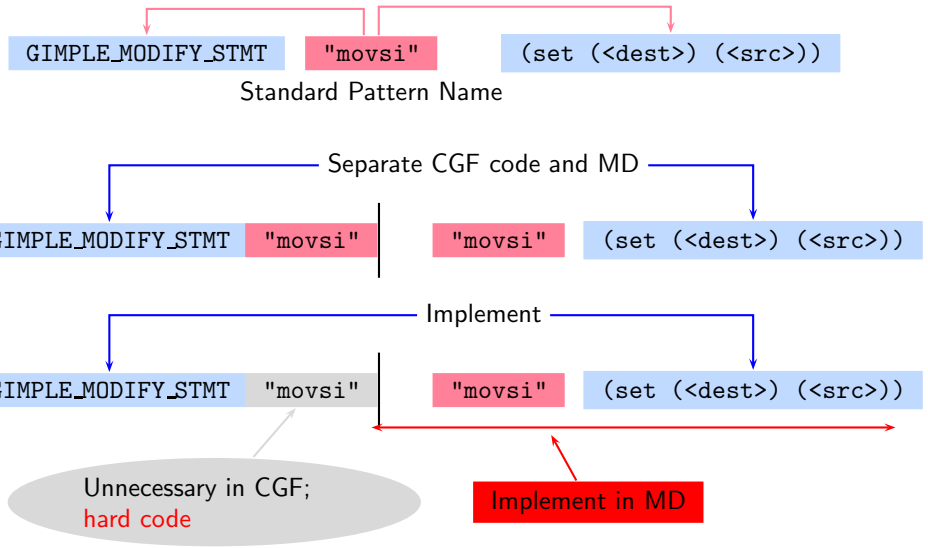
Information supplied by the MD

Notes

Part 2

Generating the Code Generators

How GCC uses target specific RTL as IR



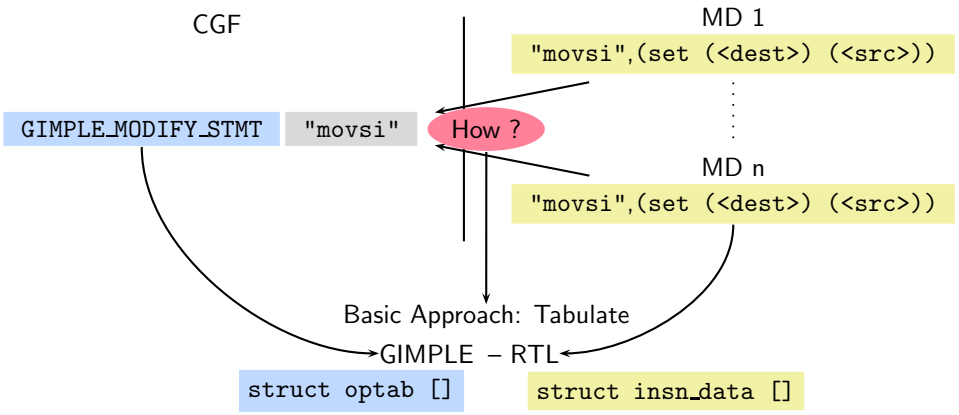
Notes

July 09 Retargetability Model: Generating the Code Generators 6/15

How GCC uses target specific RTL as IR

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Retargetability ⇒ Multiple MD vs. One CGF!



CGF needs:

An interface immune to MD authoring variations

Notes

July 09 Retargetability Model: Generating the Code Generators 7/15

Retargetability ⇒ Multiple MD vs. One CGF!

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

MD Authoring Immune Tabulation

- List insns as they appear in the chosen MD
- Index them
- Supply index to the CGF

Note
An SPN may be written at any suitable place for a given MD

July 09 Retargetability Model: Generating the Code Generators 8/15

MD Authoring Immune Tabulation

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

MD Information Data Structures

Two principal data structures

- `struct optab` – Interface to CGF
- `struct insn_data` – All information about a pattern
  - Array of each pattern read
  - Some patterns are SPNs
  - Each pattern is accessed using the generated index

Supporting data structures

- `enum insn_code`: Index of patterns available in the given MD

Note
Data structures are named in the CGF, but populated at build time.
Generating target specific code = populating these data structures.

July 09 Retargetability Model: Generating the Code Generators 9/15

MD Information Data Structures

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Assume movsi is supported but movsf is not supported...

$(SOURCE)/gcc/optabs.h
$(SOURCE)/gcc/optabs.c

optab_table

...	...
OTI_mov	mov_optab

handler

SI insn_code

SF insn_code

Notes

July 09 Retargetability Model: Generating the Code Generators 10/15

Assume movsi is supported but movsf is not supported...

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Notes

July 09 Retargetability Model: Generating the Code Generators 10/15

Assume movsi is supported but movsf is not supported...

$(SOURCE)/gcc/optabs.h
$(SOURCE)/gcc/optabs.c

optab_table

...	...
OTI_mov	mov_optab

handler

SI insn_code

SF insn_code

$(BUILD)/gcc/insn-output.c

insn_data

...	...
1280	"movsi" ... gen_movsi ...

$BUILD/gcc/insn-codes.h

```
CODE_FOR_movsi=1280
CODE_FOR_movsf=CODE_FOR_nothing
```

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

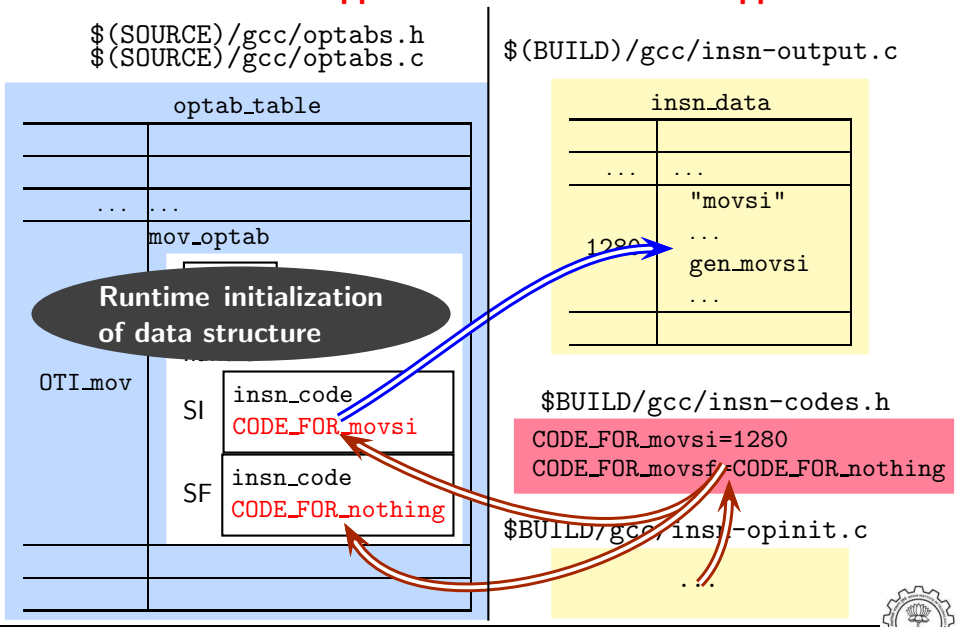
Notes

July 09 Retargetability Model: Generating the Code Generators 10/15

Assume movsi is supported but movsf is not supported...

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Assume movsi is supported but movsf is not supported...



Notes

July 09 Retargetability Model: Generating the Code Generators 10/15

Assume movsi is supported but movsf is not supported...

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

GCC Generation Phase – Revisited

Generator	Generated from MD	Information	Description
genopinit	insn-opinit.c	`void init_all_optabs (void);`	Operations Table Initialiser
gencodes	insn-codes.h	`enum insn_code = {... CODE_FOR_movsi = 1280, ...}`	Index of patterns
genooutput	insn-output.c	`struct insn_data [CODE].genfun = /* fn ptr */`	All insn data e.g. gen function
genemit	insn-emit.c	`rtx gen_rtx_movsi (/* args */) {/* body */}`	RTL emission functions

Notes

July 09 Retargetability Model: Generating the Code Generators 11/15

GCC Generation Phase – Revisited

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Explicit Calls to gen<SPN> functions

- In some cases, an entry is not made in insn_data table for some SPNs.
- gen functions for such SPNs are explicitly called.
- These are mostly related to
  - Function calls
  - Setting up of activation records
  - Non-local jumps
  - etc. (i.e. deeper study is required on this aspect)

July 09 Retargetability Model: Generating the Code Generators 12/15

Explicit Calls to gen<SPN> functions

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

Handling C Code in define_expand

```
(define_expand "movsi"
   [( set (op0) (op1))]
   ""
   "{ /* C CODE OF DEFINE EXPAND */ }")

rtx
gen_movsi (rtx operand0, rtx operand1)
{
   ...
   {
      /* C CODE OF DEFINE EXPAND */
   }
   emit_insn (gen_rtx_ SET (VOIDmode, operand0, operand1)
   ...
}
```

July 09 Retargetability Model: Generating the Code Generators 13/15

Handling C Code in define_expand

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

RTL Generation – The Internals

```
case GIMPLE_MODIFY_STMT: ... expand_assignment (...);
   ... /* Various cases of expansion */
/* One case: integer mode move */
icode = mov_optab->handler[SImode].insn_code
if (icode != CODE_FOR_nothing) {
   ... /* preparatory code */
   emit_insn (GEN_FCN(icode)(dest,src));
}
```


Part 3

Using the Code Generators

July 09 Retargetability Model: Using the Code Generators 14/15

RTL Generation – The Internals

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay

RTL to ASM Conversion

- Simple pattern matching of IR RTLs and the patterns present in all named, un-named, standard, non-standard patterns defined using `define_expand`.
- A DFA (deterministic finite automaton) is constructed and the first match is used.

July 09 Retargetability Model: Using the Code Generators 15/15

RTL to ASM Conversion

Notes

Essential Abstrations in GCC GCC Resource Center, IIT Bombay