gcc 中 RISC-V 的机器描述文件介绍

陈逸轩
wechat:XYenChi
github:https://github.com/XYenChi
blog:https://xyenchi.github.io/

参考书目:《深入分析GCC》
源码仓库:https://github.com/gcc-mirror/gcc

GCC 编译过程代码生成

机器无关

源代码(source code) → 抽象语法树(AST)/GENERIC → GIMPLE → RTL → 汇编

GCC 移植的基本步骤

1. 通过 GCC 提供的后端移植接口加入新处理器的机器描述文件。
2. 在 GCC 的编译配置文件中增加新处理器的注册信息。

机器描述文件

1. 机器描述:定义指令模板、常量、属性、断言、约束、枚举器、流水线和优化信息

   gcc/gcc/config/riscv/riscv.md

   gcc/gcc/config/riscv/constraints.md

   gcc/gcc/config/riscv/predicates.md

   gcc/gcc/config/riscv/iterator.md

   gcc/gcc/config/riscv/peephole.md
2. 目标机器相关的宏定义

   gcc/config/riscv/riscv.h
3. 目标机器相关的函数实现等

   gcc/gcc/config/riscv/riscv.cc

指令模板（Insn Pattern）定义

```
(define_expand "addsi3"
 [(set (match_operand:SI          0 "register_operand" "=r,r")
 (plus:SI (match_operand:SI 1 "register_operand" " r,r")
    (match_operand:SI 2 "arith_operand"    " r,I")))]
 ""
{
 if (TARGET_64BIT)
   {
     rtx t = gen_reg_rtx (DImode);
     emit_insn (gen_addsi3_extended (t, operands[1],
operands[2]));
     t = gen_lowpart (SImode, t);
     SUBREG_PROMOTED_VAR_P (t) = 1;
     SUBREG_PROMOTED_SET (t, SRP_SIGNED);
     emit_move_insn (operands[0], t);
     DONE;
   }
})
```

```
(define_insn "add<mode>3"
 [(set (match_operand:ANYF            0 "register_operand" "=f")
 (plus:ANYF (match_operand:ANYF 1 "register_operand" " f")
     (match_operand:ANYF 2 "register_operand" " f")))]
 "TARGET_HARD_FLOAT || TARGET_ZFINX"
 "fadd.<fmt>\t%0,%1,%2"
 [(set_attr "type" "fadd")
  (set_attr "mode" "<UNITMODE>")])


 (define_split
   [(set (match_operand:GPR 0 "register_operand")
            (match_operand:GPR 1
 "splittable_const_int_operand"))
    (clobber (match_operand:GPR 2 "register_operand"))]
   ""
   [(const_int 0)]
 {
   riscv_move_integer (operands[2], operands[0], INTVAL
 (operands[1]),
                                <GPR:MODE>mode);
   DONE;
 })
```

```
(define_insn_and_split "*zero_extendsidi2_internal"
  [(set (match_operand:DI     0 "register_operand"     "=r,r")
             (zero_extend:DI
                (match_operand:SI 1 "nonimmediate_operand" " r,m")))]
   "TARGET_64BIT && !TARGET_ZBA && !TARGET_XTHEADBB &&
!TARGET_XTHEADMEMIDX
    && !(register_operand (operands[1], SImode)
        && reg_or_subregno (operands[1]) == VL_REGNUM)"
  "@
   #
   lwu\t%0,%1"
  "&& reload_completed
   && REG_P (operands[1])
   && !paradoxical_subreg_p (operands[0])"
  [(set (match_dup 0)
             (ashift:DI (match_dup 1) (const_int 32)))
   (set (match_dup 0)
             (lshiftrt:DI (match_dup 0) (const_int 32)))]
  { operands[1] = gen_lowpart (DImode, operands[1]); }
  [(set_attr "move_type" "shift_shift,load")
   (set_attr "type" "load")
   (set_attr "mode" "DI")])
```

常量（Constant）定义

```
(define_constants
  [(RETURN_ADDR_REGNUM        1)
   (SP_REGNUM                 2)
   (GP_REGNUM                 3)
   (TP_REGNUM                 4)
   (T0_REGNUM                 5)
   (T1_REGNUM                 6)
   (S0_REGNUM                 8)
   (S1_REGNUM                 9)
   (A0_REGNUM                 10)
   (A1_REGNUM                 11)
   (S2_REGNUM                 18)
   (S3_REGNUM                 19)
   (S4_REGNUM                 20)
   (S5_REGNUM                 21)
   (S6_REGNUM                 22)
   (S7_REGNUM                 23)
   (S8_REGNUM                 24)
   (S9_REGNUM                 25)
   (S10_REGNUM                26)
   (S11_REGNUM                27)

   (NORMAL_RETURN         0)
   (SIBCALL_RETURN        1)
   (EXCEPTION_RETURN          2)
   (VL_REGNUM                 66)
   (VTYPE_REGNUM          67)
   (VXRM_REGNUM               68)
   (FRM_REGNUM                69)
])
```

属性（Attribute）定义

```
;; ISA attributes.
(define_attr "ext" "base,f,d,vector"
  (const_string "base"))

;; This attribute gives the length suffix for a sign- or zero-extension
;; instruction.
(define_mode_attr size [(QI "b") (HI "h")])

;; Mode attributes for loads.
(define_mode_attr load [(QI "lb") (HI "lh") (SI "lw") (DI "ld") (HF "flh") (SF "flw") (DF "fld")])
```

自定义断言（User-Defined Predicate)

gcc/gcc/config/riscv/predicates.md

```
(define_predicate "const_arith_operand"
  (and (match_code "const_int")
       (match_test "SMALL_OPERAND (INTVAL (op))")))
```

自定义约束（User-Defined Constraint)

gcc/gcc/config/riscv/constraints.md

```
(define_constraint "I"
  "An I-type 12-bit signed immediate."
  (and (match_code "const_int")
       (match_test "SMALL_OPERAND (ival)")))
```

枚举器（Iterator）

gcc/gcc/config/riscv/iterators.md

;; This mode iterator allows 32-bit and 64-bit GPR patterns to be generated

;; from the same template.

(**define_mode_iterator** GPR [SI (DI "TARGET_64BIT")])

;; This code iterator allows signed and unsigned widening multiplications

;; to use the same template.

(**define_code_iterator** any_extend [sign_extend zero_extend])

流水线（Pipeline）定义

gcc/gcc/config/riscv/sifive-7.md

```
(define_automaton "sifive_7")

(define_cpu_unit "sifive_7_A" "sifive_7")

(define_insn_reservation "sifive_7_load" 3
  (and (eq_attr "tune" "sifive_7")
       (eq_attr "type" "load"))
  "sifive_7_A")
```

窥孔优化

gcc/gcc/config/riscv/peephole.md

```
;; ZCMP
(define_peephole2
  [(set (match_operand:X 0 "a0a1_reg_operand")
        (match_operand:X 1 "zcmp_mv_sreg_operand"))
   (set (match_operand:X 2 "a0a1_reg_operand")
        (match_operand:X 3 "zcmp_mv_sreg_operand"))]
  "TARGET_ZCMP
   && (REGNO (operands[2]) != REGNO (operands[0]))"
  [(parallel [(set (match_dup 0)
                   (match_dup 1))
              (set (match_dup 2)
                   (match_dup 3))])]
)
```

指令模板

(define_insn 指令模板名称

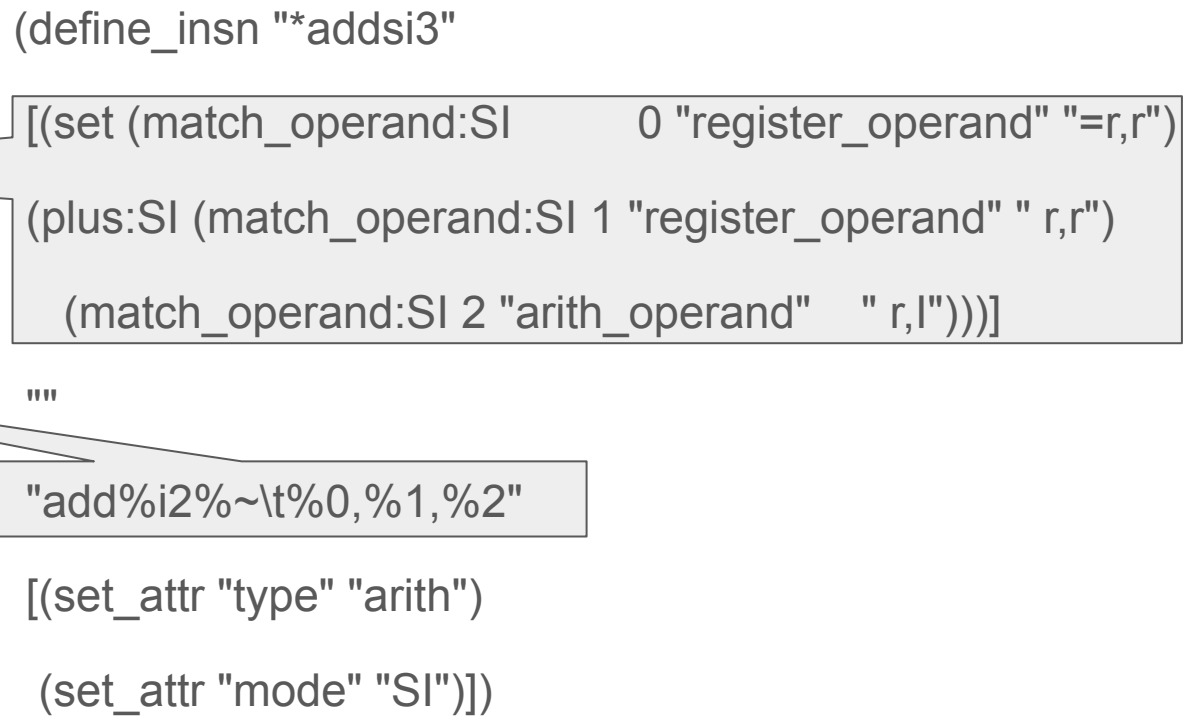
RTL 模板

条件

输出模板

属性

)



注册文件

1. 描述 CPU 类型和公司信息等。

   gcc/config.sub

2. 设置与目标机器相关的机器描述文件信息等。

   gcc/gcc/config.gcc