ACLIC Support for LLVM

zeyi2@nekoarch.cc



考古



The Original Motivation

- MCU 同时对实现面积和中断响应延迟高度敏感
- 基础 RISC-V 中断模型缺少面向低延迟 MCU 的快速中断机制

Examples

> Constrained Microcontrollers are characterized by lower performance (10s of MHz), lower memory availability (KiB), less dense technology nodes and running bare metal code. For these systems, the cycles and memory saved from fast interrupt features can have a large relative impact on the overall system cost and interrupt latency. Therefore, even features that have minimal cycle impact on interrupt latency and only bare metal code could take advantage of are considered for Constrained Microcontrollers.

https://github.com/riscv/riscv-fast-interrupt/blob/22acaa001c779ba72ccabeb6cc3fc23b7072b6cd/src/aclic.adoc?plain=1#L214-L232

SiFive E2 (2018): as few as 6 cycles to the first C-handler instruction with CLIC vectoring.

What was missing from the basic RISC-V interrupt model?

Preemption: No native same-privilege priority nesting

https://github.com/riscv/riscv-fast-interrupt/blob/62092fc6360fe22e9b3940b13310ef8658d9faff/src/clic.adoc?plain=1#L335-L348

Dispatch / Vectoring:

- Fixed local-interrupt priorities
- Standard vector slots contain jump instructions

https://github.com/riscv/riscv-fast-interrupt/blob/62092fc6360fe22e9b3940b13310ef8658d9faff/src/clic.adoc?plain=1#L373-L375

Context handling: Register save/restore remains software work

https://github.com/riscv/riscv-fast-interrupt/blob/62092fc6360fe22e9b3940b13310ef8658d9faff/src/clic.adoc?plain=1#L937-L946

在这个背景下 : ACLIC 的前身 SiFive CLIC 诞生

SiFive CLIC VS ACLIC

维度	SiFive CLIC	ACLIC
架构基础	独立的 CLIC 中断模型	基于 AIA https://github.com/riscv/riscv-aia
Priority / Nesting	独立 level + priority	复用 AIA IPRIO, 高位决定能否抢占，低位同组内排序
Vectoring	Selective HW vectoring	Smijt / Ssijt Interrupt Jump Table
Stack / Context	主要依赖软件 / 编译器	可硬件辅助 swap / push-pop

ACLIC SPEC

4 个 Fundamental ACLIC Extensions:

Interrupt domain control　　Nested Interrupt Preemption

- **`Smidctrl, Ssidctrl, Smnip, Ssnip`**

8 个 Fast Interrupt Extensions for Microcontrollers:

硬件 Vectoring　　Stack Pointer Swap　　Context Handling

- **`Smijt, Ssijt, Smehv, Ssehv, Smcsps, Sscsps, Smip, Ssip`**

LLVM Upstream



LLVM 上最早的 SiFive CLIC 支持尝试 (2020)

[RISCV] Support CLIC interruption mode named operands for CSR instructions

</> Needs Review Public https://reviews.llvm.org/D79509

[RISCV] Add SiFive's interrupt modes

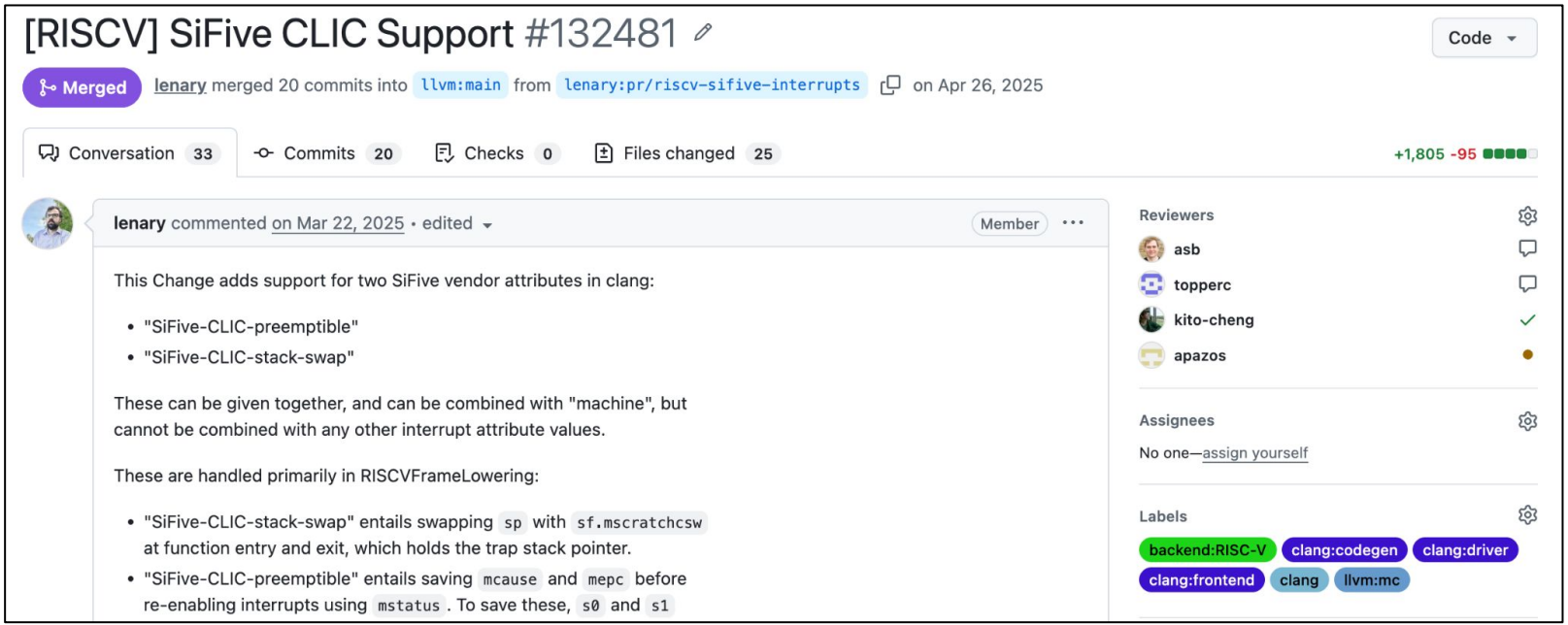
</> Needs Review Public https://reviews.llvm.org/D79521

Adds SiFive-CLIC-preemptible and SiFive-CLIC-stack-swap

没有合并的原因是:

“…at that time, no vendor extensions had been accepted upstream, and there was not yet a policy for upstream supporting vendor extensions.”

现在的 CLIC Upstream 支持 (2025)



[RISCV] SiFive CLIC Support #132481

Code

Merged lenary merged 20 commits into `llvm:main` from `lenary:pr/riscv-sifive-interrupts` on Apr 26, 2025

Conversation 33 Commits 20 Checks 0 Files changed 25 +1,805 -95

lenary commented on Mar 22, 2025 • edited Member

This Change adds support for two SiFive vendor attributes in clang:

- "SiFive-CLIC-preemptible"
- "SiFive-CLIC-stack-swap"

These can be given together, and can be combined with "machine", but cannot be combined with any other interrupt attribute values.

These are handled primarily in RISCVFrameLowering:

- "SiFive-CLIC-stack-swap" entails swapping `sp` with `sf.mscratchcsw` at function entry and exit, which holds the trap stack pointer.
- "SiFive-CLIC-preemptible" entails saving `mcause` and `mepc` before re-enabling interrupts using `mstatus`. To save these, `s0` and `s1`

Reviewers
asb
topperc
kito-cheng
apazos

Assignees
No one—assign yourself

Labels
backend:RISC-V clang:codegen clang:driver clang:frontend clang llvm:mc

当时合并版本的一些 问题

- Miscompilation: https://godbolt.org/z/439TaTx6f

The normal path can be generated without the required preemptible setup:

```
handler():
        beqz    tp, .LBB0_2
        sw      zero, 0(tp)
        mret
.LBB0_2:
        addi    sp, sp, -144
        sd      s0, 136(sp)
        sd      s1, 128(sp)
        csrr    s0, mcause
        csrr    s1, mepc
        csrsi   mstatus, 8
        ....
        call    panic()
```

As a result, the normal handler path executes and returns without ever enabling interrupt preemption. So this is a correctness issue.

当时合并版本的一些 问题

- Sema Attribute Inconsistency: https://godbolt.org/z/xG7Yc4vbh

Clang currently supports the following SiFive CLIC interrupt attribute combinations:

```
__attribute__((interrupt(
    "machine",
    "SiFive-CLIC-stack-swap")))

__attribute__((interrupt(
    "machine",
    "SiFive-CLIC-preemptible")))

__attribute__((interrupt(
    "SiFive-CLIC-stack-swap",
    "SiFive-CLIC-preemptible")))
```

However, combining all three is rejected:

https://godbolt.org/z/xG7Yc4vbh

当时合并版本的一些 问题

- Crash: https://godbolt.org/z/c6W4GYzh5

Clang enables frame pointers by default at `-O0`, while the current SiFive CLIC implementation does not support frame pointers for preemptible interrupt handlers. Instead of producing a normal diagnostic, Clang prints a crash stack trace and asks the user to report an LLVM bug.

This reproduces for both RV32 and RV64.

Reproducer: https://godbolt.org/z/c6W4GYzh5

The SiFive Interrupt Cookbook v1.2, section 3.2.4, explicitly states:

```
Using the SiFive-CLIC-preemptible attribute requires the addition of the -fomit-frame-pointer compiler flag.
```

当时合并版本的一些 问题

- Wrong Debug Info:

```
  BuildMI(MBB, MBBI, DL, TII->get(RISCV::CSRRW))
      .addReg(SPReg, RegState::Define)
      .addImm(RISCVSysReg::sf_mscratchcsw)
      .addReg(SPReg, RegState::Kill)
      .setMIFlag(FrameFlag);

  // FIXME: CFI Information for this swap.
}
```

- 不支持 Frame Pointer:

I found that will generate incorrect code when `-fno-omit-frame-pointer`, and GCC will just complain that can't compile with using frame pointer:

`x.c:8:1: error: SiFive CLIC preemptible 'interrupt' function cannot use a frame pointer`

```
$ cat x.c
void bar();

__attribute__((interrupt("SiFive-CLIC-preemptible")))
void foo()
{
  bar();
}
$ clang x.c -target riscv32 -march=rv32i_xsfmclic0p1 -menable-experimental-extensions -S -O -fno-omit-frame
...
        .type   foo,@function
foo:                                    # @foo
# %bb.0:                                # %entry
        addi    sp, sp, -80
        sw      s0, -4(s0)                      # 4-byte Folded Spill
        sw      s1, -8(s0)                      # 4-byte Folded Spill
        csrr    s0, mcause
        csrr    s1, mepc
        csrsi   mstatus, 8
        sw      ra, 68(sp)                      # 4-byte Folded Spill
        sw      t0, 64(sp)                      # 4-byte Folded Spill
...
```

Adding new ACLIC Support



MC Support Progress

Extensions	Upstream PR	Status
Smcsps, Sscsps	https://github.com/llvm/llvm-project/pull/211712	Merged
Smijt, Ssijt, Smehv, Ssehv	https://github.com/llvm/llvm-project/pull/213431	Merged
Smip, Ssip	https://github.com/llvm/llvm-project/pull/215095	Merged
Smidctrl, Ssidctrl, Smnip, Ssnip	https://github.com/llvm/llvm-project/pull/218108	In Review

Paving the way

Goal	Upstream PR	Status
Fix Sema Inconsistency Follow up: interrupt type 去重	https://github.com/llvm/llvm-project/pull/216159	Merged
Fix Crash	https://github.com/llvm/llvm-project/pull/217949	Merged
Fix Miscompilation	https://github.com/llvm/llvm-project/pull/218347	Merged
Fix Debug Info (Pt.1)	https://github.com/llvm/llvm-project/pull/218338	Merged
	Ongoing	Ongoing
Add Frame Pointer Support	https://github.com/llvm/llvm-project/pull/221318	In review

Frame Pointer Support for CLIC

