1 to many OpenRISC SMP

...

Stafford Horne

Agenda

Introduction

Implementing SMP

Introduction

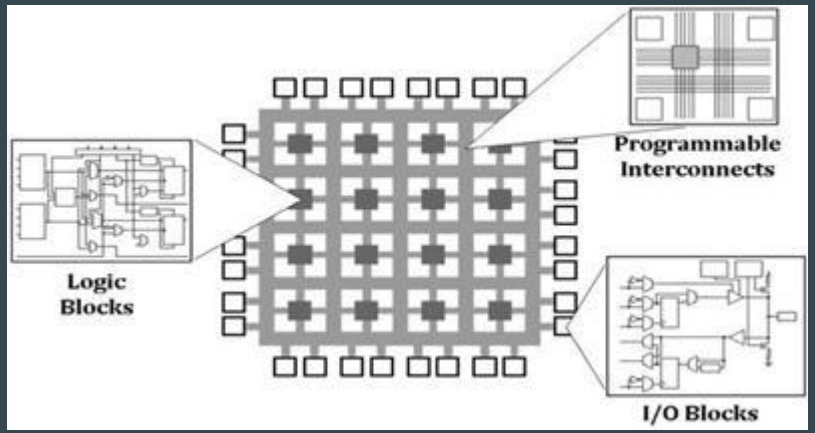
What is OpenRISC?

FPGA, IP cores

OpenCores

FuseSOC

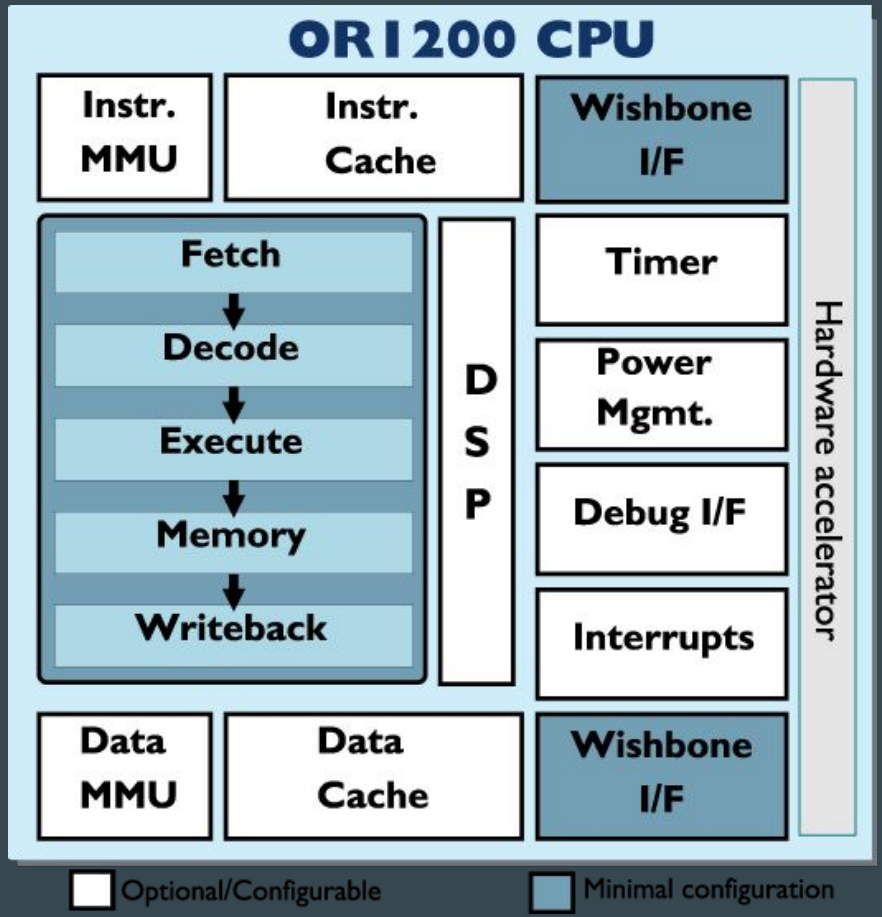
FOSSi



What is OpenRISC?

Officially OpenRISC 1000 is an open source RISC architecture:

- 32-bit / 64-bit
- 32 General Purpose Registers
- Delay Slot
- Instruction & Data MMU
- Linux support since 2010
  - 50mhz, 5 secs



Read More: https://raw.githubusercontent.com/openrisc/doc/master/openrisc-arch-1.2-rev0.pdf

OpenRISC vs Other soft codes

	Open	MMU	Arch	Linux	Silicon
OpenRISC	✓	✓	32-bit	✓	Limited
RISC-V	✓	Kinda	64-bit	4.15	Multi
Nios2	✗	✓	32-bit	✓	✓
Microblaze	✗	✓	32-bit	✓	✓

Upstream Progress

Last update in 2016 many projects were pending to go upstream

- GDB - code ok’ed pending one Copyright assignment, ETA Today
- Linux - Performance, SMP Complete
- newlib - Complete

Progress

- Qemu - Bug fixes + SMP support
- Uclibc-ng - NPTL support
- musl - no changes

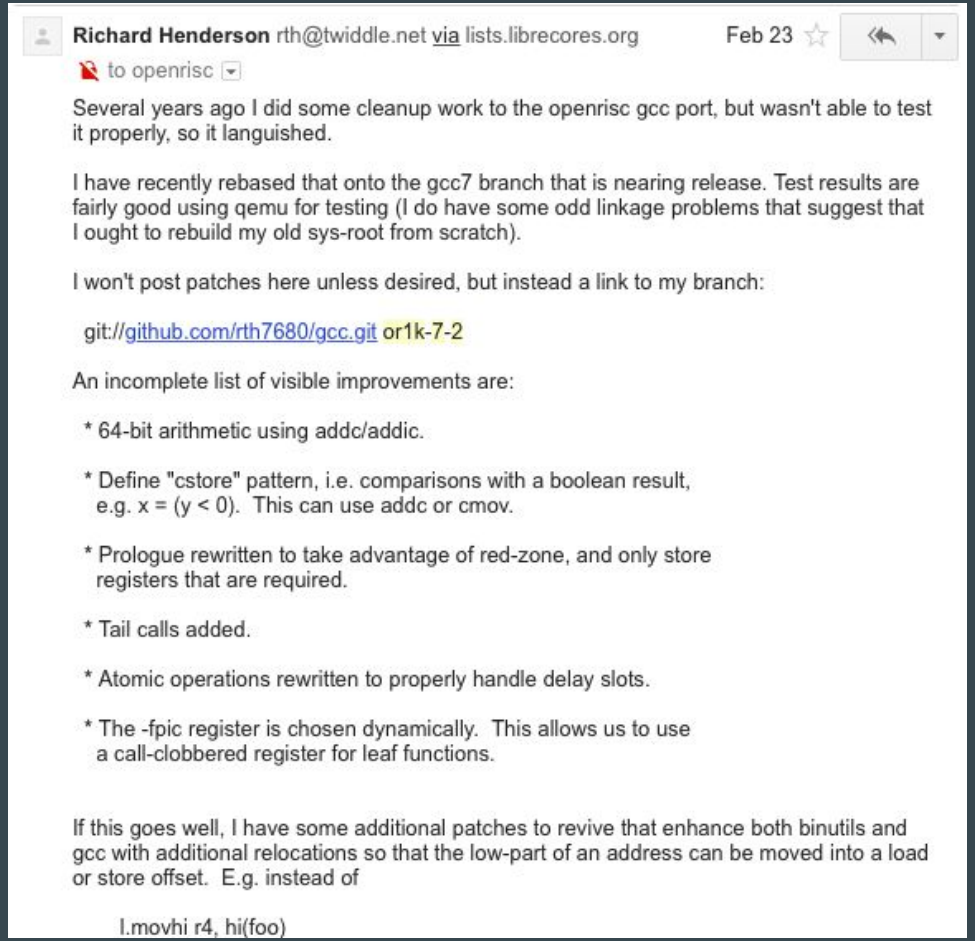
Toolchain GCC

github.com/openrisc/or1k-gcc

- 5.4.0 released (2017 Feb)

Upstream status - behind

- 8 - development (target 2018 Q1)
- 7.2 - latest 7 release (2017 Aug)
- 6.4 - latest 6 release (2016 Jul)
- 5.5 - latest 5 release (2017 Oct)



Richard Henderson rth@twiddle.net via lists.librecores.org Feb 23

to openrisc

Several years ago I did some cleanup work to the openrisc gcc port, but wasn't able to test it properly, so it languished.

I have recently rebased that onto the gcc7 branch that is nearing release. Test results are fairly good using qemu for testing (I do have some odd linkage problems that suggest that I ought to rebuild my old sys-root from scratch).

I won't post patches here unless desired, but instead a link to my branch:

git://github.com/rth7680/gcc.git or1k-7-2

An incomplete list of visible improvements are:

* 64-bit arithmetic using addc/addic.

* Define "cstore" pattern, i.e. comparisons with a boolean result, e.g. x = (y < 0). This can use addc or cmov.

* Prologue rewritten to take advantage of red-zone, and only store registers that are required.

* Tail calls added.

* Atomic operations rewritten to properly handle delay slots.

* The -fpic register is chosen dynamically. This allows us to use a call-clobbered register for leaf functions.

If this goes well, I have some additional patches to revive that enhance both binutils and gcc with additional relocations so that the low-part of an address can be moved into a load or store offset. E.g. instead of

l.movhi r4, hi(foo)

SMP

Goals

- Create a simple/low cost SMP architecture
- Learn a lot

Architecture Specification

Unique

- Interrupts are routed to every core
- Each core has a 32-bit maskable PIC
- Open Multi-core Programmable Interrupt Controller provides only IPI

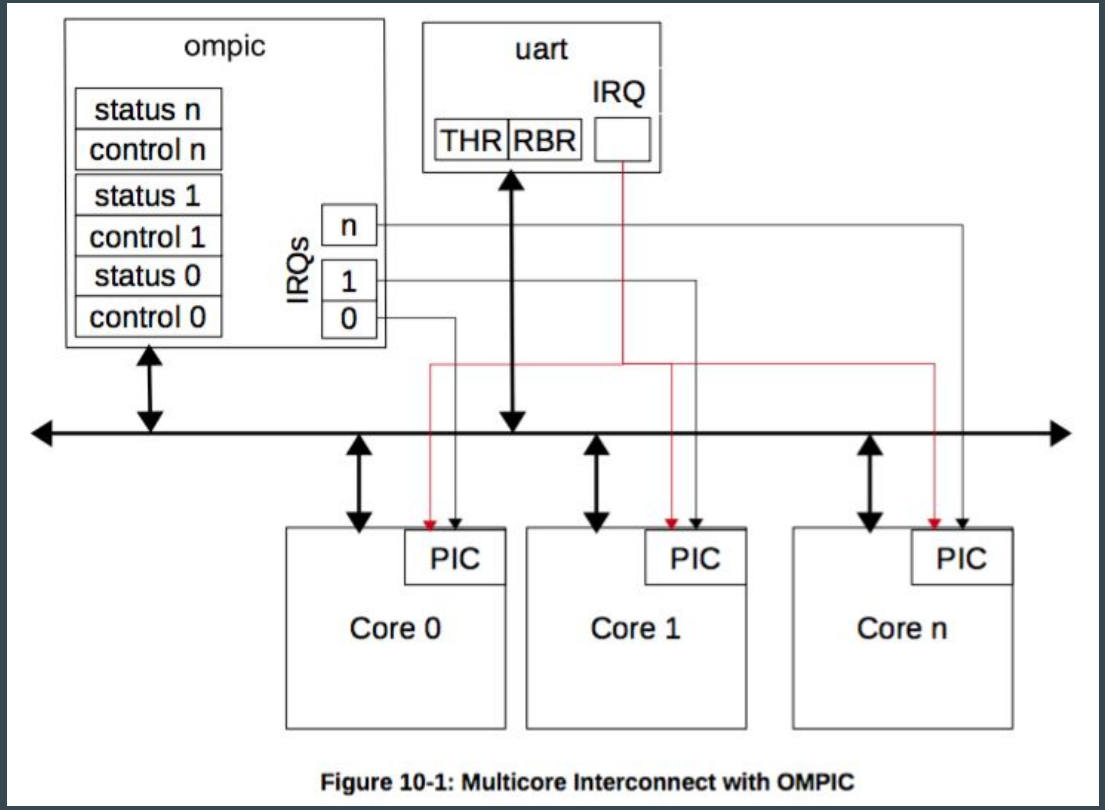
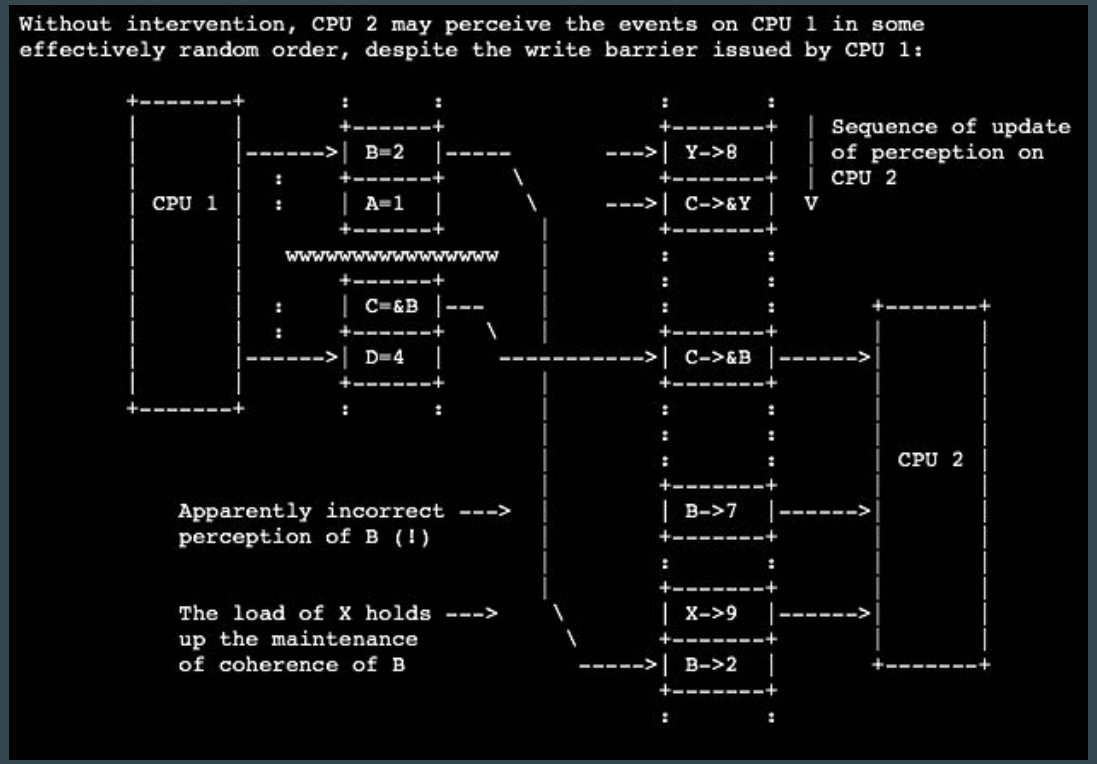


Figure 10-1: Multicore Interconnect with OMPIC

Memory Barriers

- Strong vs Weak memory model
- Memory Sync Points
- Cache Coherency
- Transitivity

OpenRISC cache snoop and atomic operations provide this.



https://www.kernel.org/doc/Documentation/memory-barriers.txt

Atomics & Spinlocks

Atomic Instructions added to support multi-core.

Peter Zijlstra recommended switching to qspinlocks and qrwlocks. Very Easy

atomic pair

```
/* Atomically perform op with v->counter and i */
#define ATOMIC_OP(op)                                                   \
static inline void atomic_##op(int i, atomic_t *v)                      \
{                                                                       \
        int tmp;                                                        \
                                                                        \
        __asm__ __volatile__(                                           \
"1:     l.lwa   %0,0(%1)        \n"                                     \
"       l." #op " %0,%0,%2      \n"                                     \
"       l.swa   0(%1),%0        \n"                                     \
"       l.bnf   1b              \n"                                     \
"        l.nop                  \n"                                     \
        : "=&r"(tmp)                                                    \
        : "r"(&v->counter), "r"(i)                                      \
        : "cc", "memory");                                              \
}
```

qspinlocks and qrwlocks

```
generic-y += percpu.h
generic-y += preempt.h
generic-y += qspinlock_types.h
generic-y += qspinlock.h
generic-y += qrwlock_types.h
generic-y += qrwlock.h
generic-y += sections.h
```

Per CPU

Provide cache aligned structure access

```
 * one-shot events, so no problem.
 */
DEFINE_PER_CPU(struct clock_event_device, clockevent_openrisc_timer);

void openrisc_clockevent_init(void)
{
        unsigned int cpu = smp_processor_id();
        struct clock_event_device *evt =
                &per_cpu(clockevent_openrisc_timer, cpu);
        struct cpuinfo_or1k *cpuinfo = &cpuinfo_or1k[cpu];

        mtspr(SPR_TTMR, SPR_TTMR_CR);

#ifdef CONFIG_SMP
        evt->broadcast = tick_broadcast;
#endif
        evt->name = "openrisc_timer_clockevent",
        evt->features = CLOCK_EVT_FEAT_ONESHOT,
        evt->rating = 300,
        evt->set_next_event = openrisc_timer_set_next_event,

        evt->cpumask = cpumask_of(cpu);

        /* We only have 28 bits */
        clockevents_config_and_register(evt, cpuinfo->clock_frequency,
```

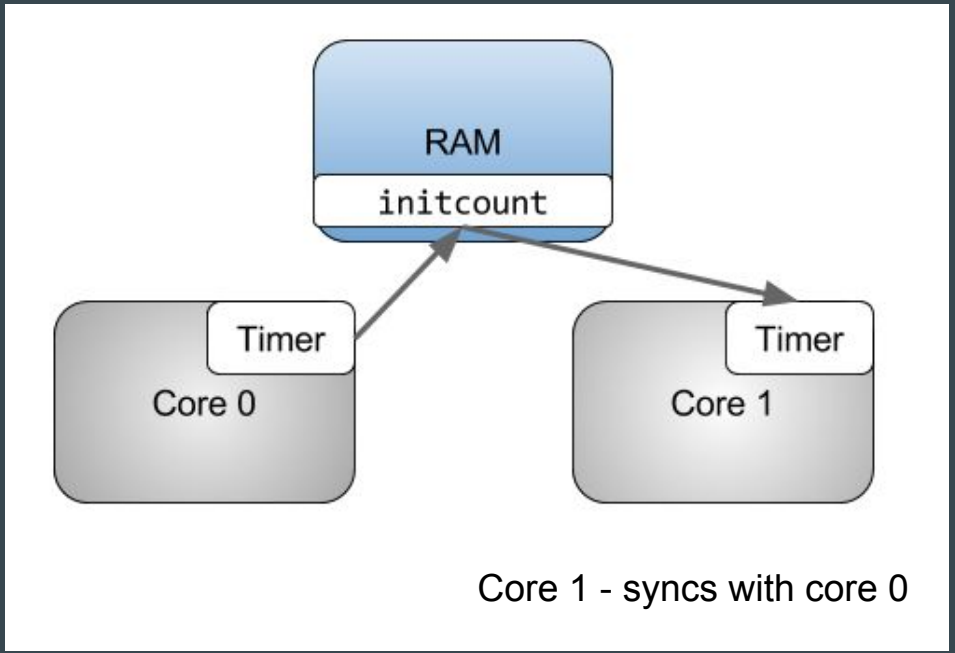
Timer Sync

Ensure each core has synchronized tick timers

Case event is handled by core 0 then later by core 1

- You don't want to go back in time
- You don't want to go into the future

CPUs must sync timer when brought online.



Lockdep and Frame Pointers

Lockdep validates locking by ensuring interrupts are on/off when locks are taken.

Requires annotating low level IRQ changes and a reliable frame unwinder.

trace_hardirqs_off when the system turns IRQ off
trace_hardirqs_on when the system turns IRQ on

Disabling Interrupts

```
930 _resume_userspace:
931         DISABLE_INTERRUPTS(r3,r4)
932         TRACE_IRQS_OFF
933         l.lwz   r4,TI_FLAGS(r10)
934         l.andi  r13,r4,_TIF_WORK_MASK
935         l.sfeqi r13,0
936         l.bf    _restore_all
937          l.nop
938
```

Re-enabling Interrupts

```
969
970 _restore_all:
971 #ifdef CONFIG_TRACE_IRQFLAGS
972         l.lwz   r4,PT_SR(r1)
973         l.andi  r3,r4,(SPR_SR_IEE|SPR_SR_TEE)
974         l.sfeq  r3,r0           /* skip trace if irqs were off */
975         l.bf    skip_hardirqs_on
976          l.nop
977         TRACE_IRQS_ON
978 skip_hardirqs_on:
979 #endif
980         RESTORE ALL
```

https://lwn.net/Articles/321663/

Patch Series

2017-11-14	Merge tag 'devicetree-for-4.15' of git://git.kernel.org/pub/scm/linux/kernel/...	Linus Torvalds	1	-2/+0
2017-11-13	Merge tag 'for-linus' of git://github.com/openrisc/linux	Linus Torvalds	39	-346/+1630
2017-11-08	kbuild: clean up *.dtb and *.dtb.S patterns from top-level Makefile	Masahiro Yamada	1	-2/+0
2017-11-03	openrisc: fix possible deadlock scenario during timer sync	Stafford Horne	1	-1/+1
2017-11-03	openrisc: pass endianness info to sparse	Luc Van Oostenryck	1	-0/+1
2017-11-03	openrisc: add tick timer multi-core sync logic	Stafford Horne	5	-3/+145
2017-11-03	openrisc: enable LOCKDEP_SUPPORT and irqflags tracing	Stafford Horne	2	-3/+74
2017-11-03	openrisc: support framepointers and STACKTRACE_SUPPORT	Stafford Horne	6	-48/+224
2017-11-03	openrisc: add simple_smp dts and defconfig for simulators	Stefan Kristiansson	2	-0/+129
2017-11-03	openrisc: add cacheflush support to fix icache aliasing	Jan Henrik Weinstock	7	-8/+194
2017-11-03	openrisc: sleep instead of spin on secondary wait	Stafford Horne	2	-2/+54
2017-11-03	openrisc: fix initial preempt state for secondary cpu tasks	Stafford Horne	2	-1/+2
2017-11-03	openrisc: initial SMP support	Stefan Kristiansson	19	-113/+584
2017-11-03	irqchip: add initial support for ompic	Stafford Horne	1	-0/+1
2017-11-03	openrisc: use qspinlocks and qrwlocks	Stafford Horne	4	-1/+24
2017-11-03	openrisc: add 1 and 2 byte cmpxchg support	Stafford Horne	1	-32/+115
2017-11-03	openrisc: use shadow registers to save regs on exception	Stefan Kristiansson	2	-26/+80
2017-11-02	License cleanup: add SPDX license identifier to uapi header files with a license	Greg Kroah-Hartman	5	-0/+5

https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/log/arch/openrisc?h=v4.15-rc1

ありがとう

thank you

On the web

me

shorne@gmail.com stffrdhrn.github.io @stffrdhrn

OpenRISC

github.com/openrisc - projects hosted here

#openrisc on freenode (I'm shorne)

openrisc@lists.librecores.org

openrisc.io

Questions

?