Booting xtensa linux on qemu

It boots:

```
parse_bootparam(phys_tag:fe000020):
Linux version 2.6.29-rc7-ga0f0129-dirty (dumb@octofox.metropolis) (gcc
version 4.4.5 (GCC) ) #7 Mon Jun 20 01:45:39 MSD 2011
bootmem_init: sysmem.bank[i:0].{type:0, start:0x1000, end:0x3000}
bootmem_init: sysmem.bank[i:1].{type:0, start:0x142000, end:0x8000000}
bootmem_init: min_low_pfn:0x1, max_low_pfn:0x8000, max_pfn:0x8000
Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 32480
Kernel command line: console=ttyS0,38400 root=/dev/simdisk0
trap_init:
PID hash table entries: 512 (order: 9, 2048 bytes)
time_init: Platform Calibrating CPU frequency

__platform_calibrate_ccount: No Platform Specific routine available to
calibrate cpu frequency!
__platform_calibrate_ccount: Using Defaults derived from kernel .config
file.
time_init: ccount_per_jiffy:100000 [10.00 MHz], nsec_per_ccount:100
console [ttyS0] enabled
Console: colour dummy device 80x25
Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)
Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)
Memory: 128380k/131072k available (865k kernel code, 2536k reserved, 53k
data, 48k init 0k highmem)
SLUB: Genslabs=12, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
calibrate_delay: Calibrating delay loop (skipped)... 10.00 BogoMIPS preset
Mount-cache hash table entries: 512
net_namespace: 520 bytes
NET: Registered protocol family 16
bio: create slab <bio-0> at 0
NET: Registered protocol family 2
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)
TCP established hash table entries: 4096 (order: 3, 32768 bytes)
TCP bind hash table entries: 4096 (order: 2, 16384 bytes)
TCP: Hash tables configured (established 4096 bind 4096)
TCP reno registered
NET: Registered protocol family 1
simdisk_init: SIMDISK: major: 240
simdisk_attach(dev:d7809000, filename:'x')
simdisk_attach: SIMDISK: disk_name:'simdisk0', filename:'x'
simdisk_attach(dev:d780941c, filename:'x')
simdisk_attach: SIMDISK: disk_name:'simdisk1', filename:'x'
msgmni has been set to 251
io scheduler noop registered (default)
loop: module loaded
Software Watchdog Timer: 0.07 initialized. soft_noboot=0 soft_margin=60 sec
```

```
(nowayout= 1)
TCP cubic registered
NET: Registered protocol family 17
ISS serial driver 0.1
VFS: Mounted root (ext2 filesystem) readonly on device 240:0.
free_initmem: Freeing unused/init kernel memory: ... 48k freed
Warning: unable to open an initial console.
Kernel panic - not syncing: No init found.  Try passing init= option to
kernel.
```

Issues

wrong address generated for jump in kernel_exception_return

arch/xtensa/kernel/entry.S:777

```
        movi    a0, 1f + (0x40000000 - 0xC0000000)      # Calculate Return
address for "1f" and store in a0
        rsil    a2, XCHAL_EXCM_LEVEL    # FIXME: again, only do this if
PS.INTLEVEL <= EXCM_LEVEL
        retw                            # rotate back by 4 registers,
possibly with underflow
                                        # Back out our _entry Frame above...

                                        # ... We return here from above
retw.
1:      mov     a1, a5                  # our a1 became a5 at the above
_entry, a1 = a5
```

Assembled it looks like this:

```
0010 ffffff3f 00000000 000000c0 05030080
...
                        1c: R_XTENSA_32 .text
...
 2fd:   000001          l32r    a0, fffc0300 <ret_from_fork+0xfffbf858>
                        2fd: R_XTENSA_SLOT0_OP  .literal+0x1c
 300:   006320          rsil    a2, 3
 303:   f01d            retw.n
 305:   051d            mov.n   a1, a5
```

Linked vmlinux looks like this (broken, word at d000357c should be 2d3d0050):

```
d0003570 ffffff3f e83b00d0 000000c0 cd3c0050
...
d0003d25:       fe1501          l32r    a0, d000357c <T$339+0x68>
d0003d28:       006320          rsil    a2, 3
d0003d2b:       f01d            retw.n
```

```
d0003d2d:       051d            mov.n   a1, a5
```

Looks like bug in linker. Cured by the following patch:

```
diff --git a/arch/xtensa/kernel/entry.S b/arch/xtensa/kernel/entry.S
index 0cb1530..e7b2263 100644
--- a/arch/xtensa/kernel/entry.S
+++ b/arch/xtensa/kernel/entry.S
@@ -774,7 +774,9 @@ _kernel_exception:
 #endif

        l32i    a3, a1, PT_PS                           # a3 = ptregs->ps
[NOTE: Used below after retw]
-       movi    a0, 1f + (0x40000000 - 0xC0000000)      # Calculate Return
address for "1f" and store in a0
+       movi    a0, 1f # + (0x40000000 - 0xC0000000)    # Calculate Return
address for "1f" and store in a0
+       movi    a2, 0x40000000 - 0xC0000000
+       add     a0, a0, a2
        rsil    a2, XCHAL_EXCM_LEVEL    # FIXME: again, only do this if
PS.INTLEVEL <= EXCM_LEVEL
        retw                            # rotate back by 4 registers,
possibly with underflow
                                        # Back out our _entry Frame above...
```

From:
http://wiki.osll.ru/ - **Open Source & Linux Lab**

Permanent link:
http://wiki.osll.ru/doku.php/etc:users:jcmvbkbc:xtensa-linux?rev=1308527373

Last update: **2011/06/20 03:49**