Supervisor Binary Interface Extension proposal

Atish Patra
Supervisor Binary Interface (SBI)

• Clear separation between Supervisor & Machine Mode
• Helps to run single OS image across different SEE
• Currently provided by Berkeley Boot Loader (BBL)

• Calling convention
  – S mode traps into M mode using **ECALL**
  – Arguments via a0-a2
  – SBI call type via a7
  – a0 is clobbered register

• Documentation available at
Current status

• Fixed
• No backward compatibility
• Not standardized
• No standard return/error value

<table>
<thead>
<tr>
<th>Type</th>
<th>Function</th>
<th>Function ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timer</td>
<td>sbi_set_timer</td>
<td>0</td>
</tr>
<tr>
<td>Console</td>
<td>sbi_console_putchar</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>sbi_console_getchar</td>
<td>2</td>
</tr>
<tr>
<td>IPI</td>
<td>sbi_clear_ipi</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>sbi_send_ipi</td>
<td>4</td>
</tr>
<tr>
<td>Memory Model</td>
<td>sbi_remote_fence_i</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>sbi_remote_sfence_vma</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>sbi_remote_sfence_vma_as</td>
<td>7</td>
</tr>
<tr>
<td>Shutdown</td>
<td>sbi_shutdown</td>
<td>8</td>
</tr>
</tbody>
</table>
SBI Extension Proposal - Objective

• Make it extendable & backward compatible
• Shouldn’t be a kitchen sink for all hardware abstractions
• Continue to remain minimalistic
• Make it modular to facilitate easy discussion/debate
  – Focus on Legacy (v0.1) & Base SBI
  – Power management/Vendor extension to follow
• Define standard error codes
  – A mapping function between SBI error code & OS specific error
• Ongoing discussion
  – V1
    • http://lists.infradead.org/pipermail/linux-riscv/2018-October/001974.html
  – V2
SBI extension proposal – Calling convention

• Function ID numbering scheme
  – Static binding
    • Bit[31:24] = Function Type Set
    • Number Bit[23:0] = Function Number within Function type
  – Dynamic binding
    • DT Based dynamic probing individual functions

• Return Values
  – Just single return value with value (>=0) & error (<0)
  – Return a struct with a value & error (ABI says both a0 & a1 are return values)
  – Error strings in per hart local area?

<table>
<thead>
<tr>
<th>Function Type</th>
<th>Function ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Function</td>
<td>0x0</td>
<td>Base APIs mandatory for any SBI version</td>
</tr>
<tr>
<td>HART PM APIs</td>
<td>0x1</td>
<td>Hart UP/Down/Suspend APIs for per-Hart power management</td>
</tr>
<tr>
<td>System PM APIs</td>
<td>0x2</td>
<td>System Shutdown/Reboot/Suspend for system-level power management</td>
</tr>
<tr>
<td>Vendor APIs</td>
<td>0xff</td>
<td>Vendor specific APIs</td>
</tr>
</tbody>
</table>
### SBI extension proposal - Scope

<table>
<thead>
<tr>
<th>SBI Extension</th>
<th>Function</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Function</td>
<td>sbi_get_version, sbi_check_api, or get_feature()</td>
<td>Mandatory</td>
</tr>
<tr>
<td>HART PM APIs</td>
<td>sbi_hart_up, sbi_hart_suspend, sbi_hart_down, sbi_hart_get_state</td>
<td>Optional</td>
</tr>
<tr>
<td>System PM APIs</td>
<td>sbi_system_shutdown, sbi_system_reset</td>
<td>Optional</td>
</tr>
<tr>
<td>Vendor APIs</td>
<td>TODO</td>
<td>Optional</td>
</tr>
</tbody>
</table>
OpenSBI project

• BBL/Coreboot provides separate SBI implementation
• More fragmentations going forward considering vendor specific usage
• Difficult to maintain & track the SBI changes
• OpenSBI to rescue!!
  – Create a open separate SBI implementation project
  – SBI implementation built as static library that any boot loader(both S mode & M mode) can use
  – Provides a reference implementation that potentially can replace BBL as well
  – This reference implementation can produce firmware binary that can be used as in M mode directly

• Recommended License ?
• Any feedback ?
Feedbacks

• u32 sbi_check_api(unsigned long start_api_id, unsigned long count) vs u32 sbi_get_version(void)

• Treat SBI function return & global error separately
  ▪ No global state should be maintained
  ▪ pass by reference
  ▪ return a struct containing err & return value (My choice!!)
  ▪ Should we change all the mandatory existing functions as well to reflect that?

• May be rename it to a generic interface instead of Supervisor Binary Interface
• Anything else?
Etrap call

- lui t0, #ecallTableBase # or auipc
- slli t1, a6, 2 #3 for 64 bit
- add t0, t0, t1
- jr, nnn(t0) # lo bits of ecallTableBase, if not 4k aligned
- # called function does the mret