Enable deepest suspend power state on S2I

Chunyan Zhang
April 2019
About me

- From Unisoc (Spreadtrum)
- Worked in KWG
- Now in PMWG
The Goal

- Current
  - S2R (suspend to ram)
  - S2I (Suspend to idle)

- Future
  - S2R
  - S2I
Dev environment

- Board
  - Unisoc's mobile phone with the processor SC9863A which has eight cores of Cortex A55 in two clusters.
- Mainline kernel with a few necessary drivers (uart, clock, gpio)
- PSCI 1.0
- Power topology
Both S2R and S2I would call `psci_cpu_suspend_start()` but with different parameters:

**S2R**
- `target_pwrlvl=1`
- `is_power_down_state=1`
- `state_info[0]=ARM_LOCAL_STATE_OFF`
- `state_info[1]=ARM_LOCAL_STATE_OFF`

**S2I**
- `target_pwrlvl=0`
- `is_power_down_state=1`
- `state_info[0]=ARM_LOCAL_STATE_OFF`
- `state_info[1]=ARM_LOCAL_STATE_RUN`
Deepest c-state

- "arm,psci-suspend-param" in device tree “idle-state”
  - Let S2R and S2I have same parameter to call psci_cpu_suspend_start()
## Power consumption on below scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Normal running</th>
<th>S2R</th>
<th>S2I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without deepest c-state</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Eight cores</td>
<td>190 ~ 205mW</td>
<td>~139mW</td>
<td>175 ~ 190mW</td>
</tr>
<tr>
<td>- Two clusters, one core in each cluster</td>
<td>190 ~205mW</td>
<td>~139mW</td>
<td>175 ~ 190mW</td>
</tr>
<tr>
<td>- One core in DT</td>
<td>181 ~196mW</td>
<td>~139mW</td>
<td>175 ~ 190mW</td>
</tr>
<tr>
<td>With deepest c-state</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- One core in each cluster; Or - Two core in one cluster;</td>
<td>190 ~205mW</td>
<td>~139mW</td>
<td>175 ~ 190mW</td>
</tr>
<tr>
<td>- Only 1 core alive</td>
<td>177 ~194mW</td>
<td>~139mW</td>
<td>~139mW</td>
</tr>
</tbody>
</table>
Plug cores when suspend

- During S2R, secondary cores would be unplugged
- During S2I, secondary cores would **NOT** be unplugged
Switch to using generic power domain

- No much change on power consumption of the scenarios listed previously for now.
DDR self refresh for S2R and S2I

- Looked on my board:
  - DDR device is set “auto self-refresh”
  - DDR granule would be set self-refresh for both suspend
  - DDR controller seems not in sleep state for both suspend (from what I’ve known)
  - Probably because that some components on the board haven’t been put into deep sleep state.
Next to go

- To find out which part of components or processors didn’t enter into deep sleep.
- To look at psci_cpu_off() further
- Any suggestions?
Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contactus@linaro.org