Close to Hardware – Driver Implementation

Peripheral Interface
Agenda

- Overview of GTA02
- Bq27k
- HDQ
- FIQ
Overview of GTA02

- G-sensor
- Bq27k
- S3C2442
- Glamo
- PMU
- AudioCodec
- GPS
- GSM
- BT
- WiFi
- UART
- I²C
- SPI
- SDIO
- DP/DN
Interface: HDQ

- G-sensor
- Bq27k
- S3C2442
- WiFi
- Glamo
- AudioCodec
- PMU
- GPS
- GSM
- BT
- UART
- SPI
- SDIO
- DP/DN
- I²C
- MEMORY BUS
- HDQ
Driver Implementation

• First three things
  – 1. Read datasheet
  – 2. Read datasheet
  – 3. Read datasheet again

• Familiar with peripheral interfaces
  – Linux API is a good reference

• Good tools are prerequisite to the successful execution of a job
  – Scope
  – Logic Analyzer
  – Memter
Bq27k

• Bq27k is
  – Battery capacity monitoring and reporting device

• Why we use Bq27k?
  – Space consideration?
  – Something new can be hacked?

• What’s the interface of this device?
  – HDQ
  – I²C

• Why HDQ?
Typical Application
HDQ Communication Basics

• HDQ means??
• HDQ is a protocol
  – Asynchronous
  – Return-to-one
  – Single-wire
  – Open Drain
• Implementing the HDQ interface
  – Discrete I/O
  – UART
Typical HDQ read cycle
Break and Break Recovery
Host/Slave Transmitted Bit
### Timing Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Device</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{(B)}$</td>
<td>All</td>
<td>190 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{(BR)}$</td>
<td>All</td>
<td>40 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{\text{start-detect}}$</td>
<td>bq2019, bq262x0</td>
<td>5 ns (1)</td>
<td></td>
</tr>
<tr>
<td>$t_{(HW1)}$</td>
<td>bq2019, bq262x0</td>
<td>32 μs (1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bq26500</td>
<td>17 μs (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bq26051, bq270x0</td>
<td>0.5 μs (2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>50 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{(HW0)}$</td>
<td>bq2019, bq262x0, bq26500</td>
<td>100 μs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bq26501, bq270x0</td>
<td>86 μs (3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>145 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{(CYCH)}$</td>
<td>All</td>
<td>190 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{(DW1)}$</td>
<td>All</td>
<td>32 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{(DW0)}$</td>
<td>All</td>
<td>80 μs</td>
<td></td>
</tr>
<tr>
<td>$t_{(CYCD)}$</td>
<td>All but bq2650x</td>
<td>190 μs</td>
<td>250 μs</td>
</tr>
<tr>
<td></td>
<td>bq2650x</td>
<td>260 μs (4)</td>
<td></td>
</tr>
<tr>
<td>$t_{(RSPS)}$</td>
<td>All</td>
<td>190 μs</td>
<td>320 μs</td>
</tr>
</tbody>
</table>
The real world of HDQ
Driver Implementation of HDQ

- Interrupts Using I/O Port for HDQ
- We need a TIMER Interrupt
- What’s the timing resolution?
  - 32us
- How to make sure timing is correct?
- UART interface to HDQ
- Interrupt do not affect the timing of HDQ communication
HDQ Communication with UART
FIQ

- FIQ - Fast Interrupt Request
- Why FIQ?
  - The priority of FIQ is higher than IRQ
  - Atomic operation
- The vector of FIQ is 0x18