ChipIdea Highspeed Dual Role Controller Driver

1. How to test OTG FSM(HNP and SRP)

To show how to demo OTG HNP and SRP functions via sys input files with 2 Freescale i.MX6Q sabre SD boards.

1.1 How to enable OTG FSM

1.1.1 Select CONFIG_USB_OTG_FSM in menuconfig, rebuild kernel

Image and modules. If you want to check some internal variables for otg fsm, mount debugfs, there are 2 files which can show otg fsm variables and some controller registers value:

```
cat /sys/kernel/debug/ci_hdrc.0/otg
cat /sys/kernel/debug/ci_hdrc.0/registers
```

1.1.2 Add below entries in your dts file for your controller node

```
otg-rev = <0x0200>;
adp-disable;
```

1.2 Test operations

1. Power up 2 Freescale i.MX6Q sabre SD boards with gadget class driver loaded (e.g. g_mass_storage).

2. Connect 2 boards with usb cable with one end is micro A plug, the other end is micro B plug.

    The A-device(with micro A plug inserted) should enumerate B-device.

3. Role switch

    On B-device:

    ```
    echo 1 > /sys/bus/platform/devices/ci_hdrc.0/inputs/b_bus_req
    ```

    B-device should take host role and enumerate A-device.

4. A-device switch back to host.

    On B-device:

    ```
    echo 0 > /sys/bus/platform/devices/ci_hdrc.0/inputs/b_bus_req
    ```

    or, by introducing HNP polling, B-Host can know when A-peripheral wish to be host role, so this role switch also can be trigged in A-peripheral side by answering the polling from B-Host, this can be done on A-device:

    ```
    echo 1 > /sys/bus/platform/devices/ci_hdrc.0/inputs/a_bus_req
    ```

    A-device should switch back to host and enumerate B-device.

5. Remove B-device(unplug micro B plug) and insert again in 10 seconds, A-device should enumerate B-device again.

6. Remove B-device(unplug micro B plug) and insert again after 10 seconds, A-device should NOT enumerate B-device.

    if A-device wants to use bus:

    On A-device:

    ```
    echo 0 > /sys/bus/platform/devices/ci_hdrc.0/inputs/a_bus_drop
    echo 1 > /sys/bus/platform/devices/ci_hdrc.0/inputs/a_bus_req
    ```

    if B-device wants to use bus:

    On B-device:

    ```
    echo 1 > /sys/bus/platform/devices/ci_hdrc.0/inputs/b_bus_req
    ```

7. A-device power down the bus.

    On A-device:

    ```
    echo 1 > /sys/bus/platform/devices/ci_hdrc.0/inputs/a_bus_drop
    ```

    A-device should disconnect with B-device and power down the bus.

8. B-device does data pulse for SRP.

    On B-device:

```
echo 1 > /sys/bus/platform/devices/ci_hdrc.0/inputs/b_bus_req
```

A-device should resume usb bus and enumerate B-device.

1.3 Reference document

"On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification July 27, 2012 Revision 2.0 version 1.1a"

2. How to enable USB as system wakeup source

Below is the example for how to enable USB as system wakeup source at imx6 platform.

2.1 Enable core's wakeup:

```
echo enabled > /sys/bus/platform/devices/ci_hdrc.0/power/wakeup
```

2.2 Enable glue layer's wakeup:

```
echo enabled > /sys/bus/platform/devices/2184000.usb/power/wakeup
```

2.3 Enable PHY's wakeup (optional):

```
echo enabled > /sys/bus/platform/devices/20c9000.usbphy/power/wakeup
```

2.4 Enable roothub's wakeup:

```
echo enabled > /sys/bus/usb/devices/usb1/power/wakeup
```

2.5 Enable related device's wakeup:

```
echo enabled > /sys/bus/usb/devices/1-1/power/wakeup
```

If the system has only one usb port, and you want usb wakeup at this port, you can use below script to enable usb wakeup:

```
for i in $(find /sys -name wakeup | grep usb);do echo enabled > $i;done;
```