Contents

[1. Introduction: - 2](#_Toc122968998)

[2. PCIe platform/ Controller driver flow in Exynos: - 2](#_Toc122968999)

[2.1. PCI core code flow: - 2](#_Toc122969000)

[2.1.1. pci\_device\_probe(): - 2](#_Toc122969001)

[3. PCIe device (Cypress WLAN) code flow: - 3](#_Toc122969002)

[a. PCIe enumeration: - 3](#_Toc122969003)

# Introduction: -

**PCI enumeration: -**

The OS can discover them via the standard PCI enumeration mechanism, using config accesses to discover and identify devices and read and size their BARs.

# PCIe platform/ Controller driver flow in Exynos: -

File - <Android\_Root\_Folder>/Android-kernel/exynos/drivers/pci/controller/dwc/v920/ pcie-exynos-v920-rc-auto.c

struct platform\_driver exynos\_pcie\_driver = {

.probe = exynos\_v920\_pcie\_probe,

…

};

static int pcie\_init\_v920(void)

{

return platform\_driver\_register(&exynos\_pcie\_driver);

}

## 2.1. PCI core code flow: -

### 2.1.1. pci\_device\_probe(): -

File: - Android-kernel/exynos/drivers/pci/pci\_driver.c

struct bus\_type pci\_bus\_type = {

.name = "pci",

.match = pci\_bus\_match,

.uevent = pci\_uevent,

.probe = pci\_device\_probe,

….

}

pci\_device\_probe()

{

1. pci\_assign\_irq( );
2. pci\_dev\_get( );
3. \_\_pci\_device\_probe( );

{

pci\_match\_device( );

pci\_call\_probe( ) {

local\_pci\_probe () {

rc = pci\_drv->probe(pci\_dev, ddi->id);

|= = > Probably this function calls drivers probe function.

}

}

}

}

pci\_dev\_get(): -

Increments the reference count of the pci device structure, when they bind to a device, and release them by calling pci\_dev\_put(), in their disconnect() methods.

\_\_pci\_device\_probe ( ) : –

Check if a driver wants to claim a specific PCI device.

@ drv : driver to call to check if it wants the PCI device

@ pci\_dev : PCI device being probed

pci\_match\_device ( ): -

See if a device matches a driver's list of IDs

@ drv: the PCI driver to match against

@ dev: the PCI device structure to match against

# PCIe device (Cypress WLAN) code flow: -

Reference - Android\_Code\_Sync\_13122022/Android-kernel/exynos/Documentation/PCI/pci.rst

PCI drivers "discover" PCI devices in a system via pci\_register\_driver(). Actually, it's the other way around. When the PCI generic code discovers a new device, the driver with a matching "description" will be notified.

## PCIe enumeration: -

Function - dhdpcie\_init()

File - Android-kernel/exynos/drivers/net/wireless/cypress/bcmdhd/dhd\_pcie\_linux.c

int dhdpcie\_init(struct pci\_dev \*pdev)

{

….

/\* Find the PCI resources, verify the \*/

/\* vendor and device ID, map BAR regions and irq, update in structures \*/

if (dhdpcie\_scan\_resource(dhdpcie\_info)) {

DHD\_ERROR(("%s: dhd\_Scan\_PCI\_Res failed\n", \_\_FUNCTION\_\_));

break;

}

….

}