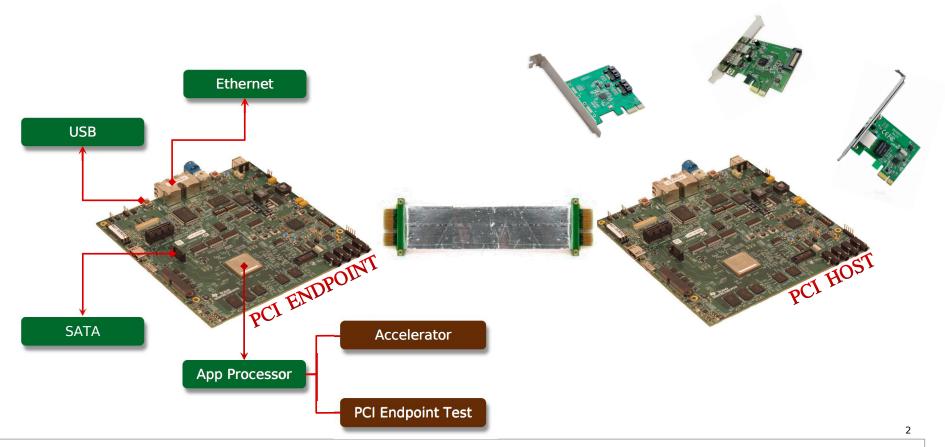
Linux PCI EP Framework

Support for Configurable PCI Endpoint in Linux

KISHON VIJAY ABRAHAM I



Introduction



EP Responsibilities write header PCI EndpointTest magic command PCI STANDARD HEADER status deviceid vendorid src addr status command dst addr classcode revid size BIST hdrtype lattimer. cache... checksum BAR0 BAR1 USB capability registers BAR2 operational registers BAR3 runtime registers BAR4 doorbell array BAR5 cardbus CIS pointer SATA subsys id subsys vendor id Generic host control operational registers expansion ROM base address vendor specific reserved cap.ptr. port 0 reserved

map BAR

port 31

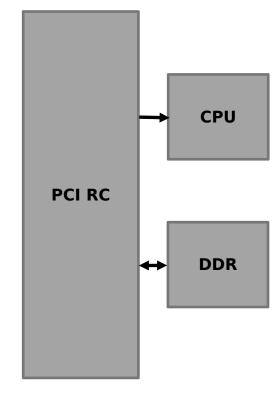
min.lat

int.pin

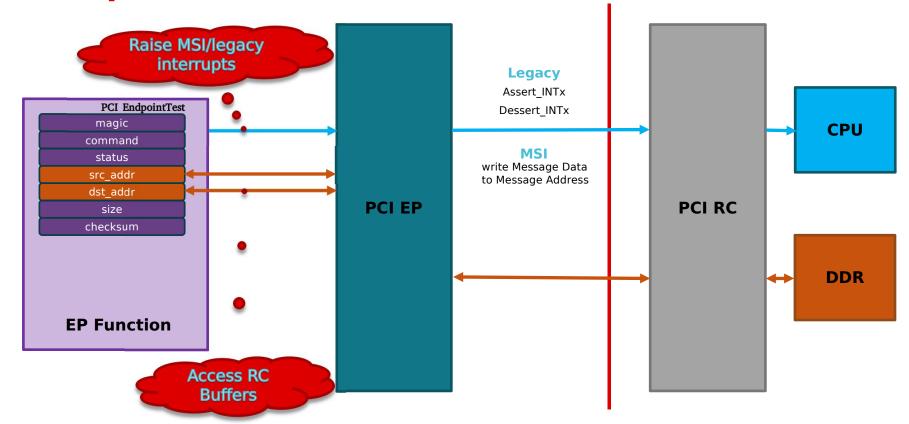
PCI Endpoint

min.gnt

int.line



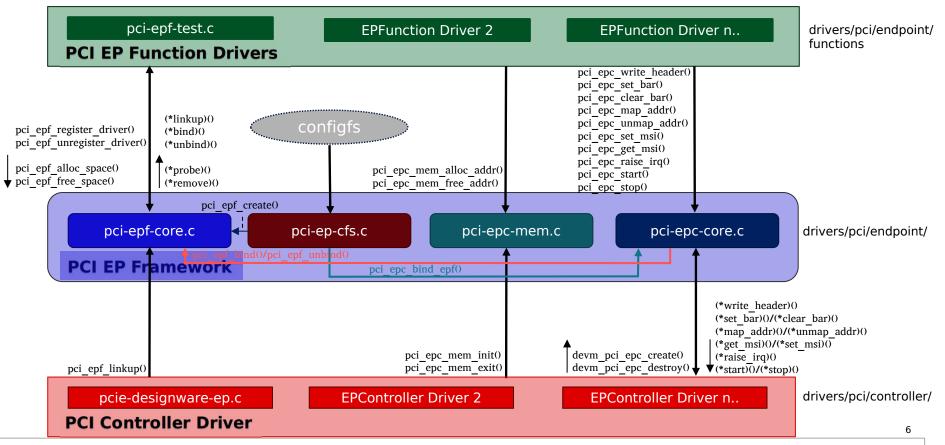
EP Responsibilities



EP Framework

struct pci epc struct pci epf struct pci epc ops struct pci epf driver (*write header)(..) struct device dev: struct device dev; int (*probe)(struct pci epf *epf); const struct pci epc ops *ops; (*set bar)(..) const char *name; int (*remove)(struct pci epf *epf); struct pci epf header *header; struct pci epc mem *mem; (*clear bar)(..) struct device driver driver; spinlock t lock; (*map addr)(..) struct pci epf bar bar[6]; const struct pci epf device id *id table; (*unmap_addr)(..) u8 msi interrupts; struct pci epc *epc; (*get_msi)(..) (*set msi)(..) struct pci epf driver *driver; (*raise irq)(..) const char *pci epc name; (*start)(..) (*stop)(..) struct pci epf header struct pci epc mem struct pci epf ops u16 vendorid: u16 deviceid; int (*bind)(struct pci epf *epf); phys addr t phys base; u8 revid: void (*unbind)(struct pci epf *epf); unsigned long *bitmap; u8 progif code; void (*linkup)(struct pci epf *epf); int pages; u8 subclass code; u8 baseclass code; u8 cache line size; u16 subsys vendor id; u16 subsys id; enum pci interrupt pin interrupt pin;

EP Framework





devicetree node

```
pcie1 rc: pcie rc@51000000 {
                                                                             pcie1 ep: pcie ep@51000000 {
  compatible = "ti,dra7-pcie";
  reg = \langle 0x51000000 \ 0x1000 \rangle, \langle 0x51002000 \ 0x14c \rangle, \langle 0x1000 \ 0x2000 \rangle;
  reg-names = "rc dbics", "ti conf", "config";
  interrupts = <0 232 0x4>, <0 233 0x4>;
  ti,hwmods = "pcie1";
                                                                               interrupts = <0 232 0x4>;
  phys = < &pcie1 phy>;
                                                                               ti,hwmods = "pcie1";
  phy-names = "pcie-phy0";
                                                                               phys = <&pcie1 phy>;
                                                                               phy-names = "pcie-phy0";
  num-lanes = <1>;
  \#address-cells = <3>;
                                                                               num-lanes = <1>;
                                                                               num-ib-windows = <4>;
  \#size-cells = <2>;
  device type = "pci";
                                                                               num-ob-windows = <16>;
  ranges = <0x81000000 0 0
                                         0x03000 0 0x00010000
             0x82000000 0 0x20013000 0x13000 0 0xffed000>;
   #interrupt-cells = <1>;
  linux,pci-domain = <0>;
  interrupt-map-mask = <0 \ 0 \ 0 \ 7>;
  interrupt-map = <0 \ 0 \ 0 \ 1 &pcie1 intc 1>,
};
                  RC Devicetree Node
```

```
compatible = "ti,dra7-pcie-ep";
reg = \langle 0x51000000 \ 0x1000 \rangle, \langle 0x51002000 \ 0x14c \rangle,
      <0x51001000 0x80>, <0x1000 0xFFFF000>;
reg-names = "ep dbics", "ti conf", "ep dbics2", "addr space";
syscon-legacy-mode = <8scm conf1 0x14 2>;
```

EP Devicetree Node

Configuring PCI Endpoint Function

```
# ls /sys/class/pci epc/
                                                                        list of PCI Endpoint Controllers
          51000000.pcie ep
# ls /sys/bus/pci-epf/drivers
          pci epf test
                                                                 list of PCI Endpoint Function Drivers
# mount -t configfs none /sys/kernel/config
# cd /sys/kernel/config/pci ep/
# mkdir pci epf test.0
# cd pci epf test.0
# 1s
          baseclass code
                          function
                                                        vendorid
                                        revid
          cache line size
                          interrupt pin
                                        subclass code
                          peripherāl
          deviceid _
                                        subsys Id
                          progif code
                                        subsys vendor id
          epc
# cat vendorid
          0xffff
# cat interrupt pin
          0x0001
                                                                   Creating Endpoint Function Device
  echo 0x104c > vendorid
# echo 16 > msi interrupts
                                                               Configuring Endpoint Function Device
# echo "51000000.pcie_ep" > epc
                                              Binding Endpoint Function with Endpoint Controller
```

Happy Hacking!

Feedback:

kishon@ti.com kishonvijayabraham@gmail.com

