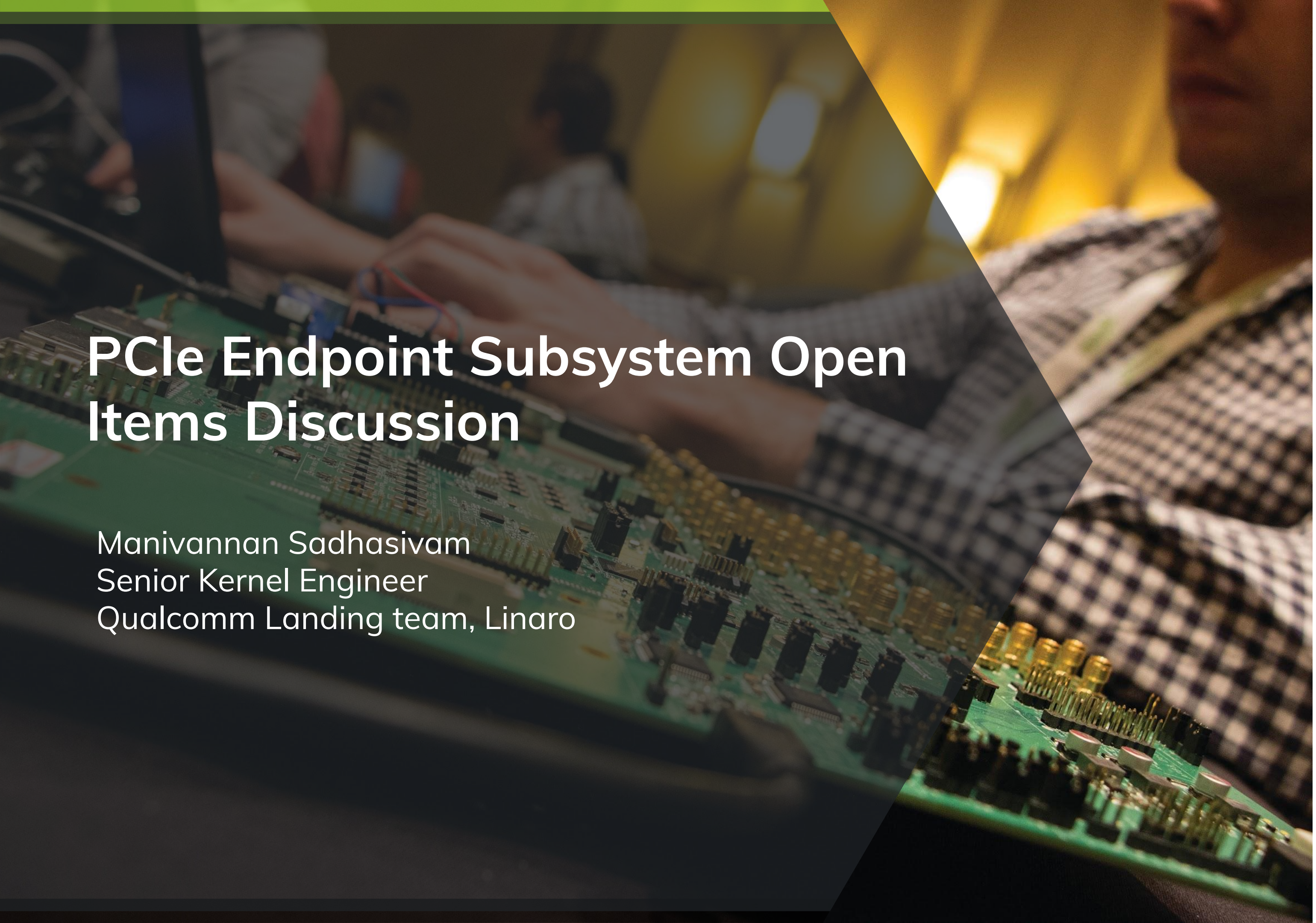


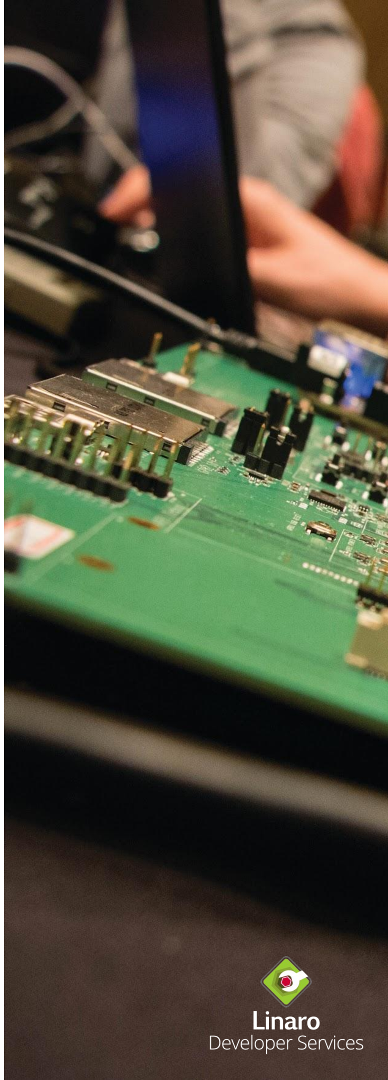
# PCIe Endpoint Subsystem Open Items Discussion

Manivannan Sadhasivam
Senior Kernel Engineer
Qualcomm Landing team, Linaro

Linaro
Developer Services

# Topics

- Reworking PCIe Endpoint notification
- Devicetree Integration



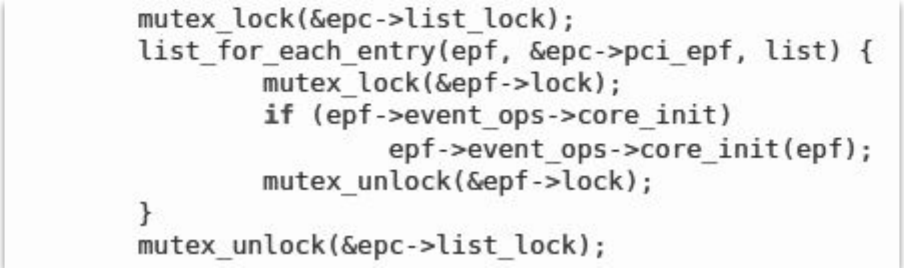
Linaro
Developer Services

# Reworking PCIe Endpoint notification

- PCIe Endpoint framework uses “atomic notifiers” for passing the events from EPC (Endpoint Controller) to EPF (Endpoint Function)
    - CORE_INIT - Signalling the initialization of EPC
    - LINK_UP - PCIe Link Up event
- This suffers from several issues:
    - The EPF notification function should be in atomic context
        - Present EPF drivers in mainline suffers from “Sleeping in atomic” bug
    - Notifiers are not a good candidate for passing the events if there is a fixed interface between sender and receiver
        - Suggested by Rob Herring

# Reworking PCIe Endpoint notification

- Proposal
  - A simple callback mechanism for passing the events from EPC to EPF
  - EPF drivers should populate the callbacks in “epf->event_ops” structure
    - core_init()
    - link_up()
  - EPC core will just execute the callbacks for each EPF driver associated with the EPC on the occurrence of the event



```
        mutex_lock(&epc->list_lock);
        list_for_each_entry(epf, &epc->pci_epf, list) {
                mutex_lock(&epf->lock);
                if (epf->event_ops->core_init)
                        epf->event_ops->core_init(epf);
                mutex_unlock(&epf->lock);
        }
        mutex_unlock(&epc->list_lock);
```

# Devicetree Integration

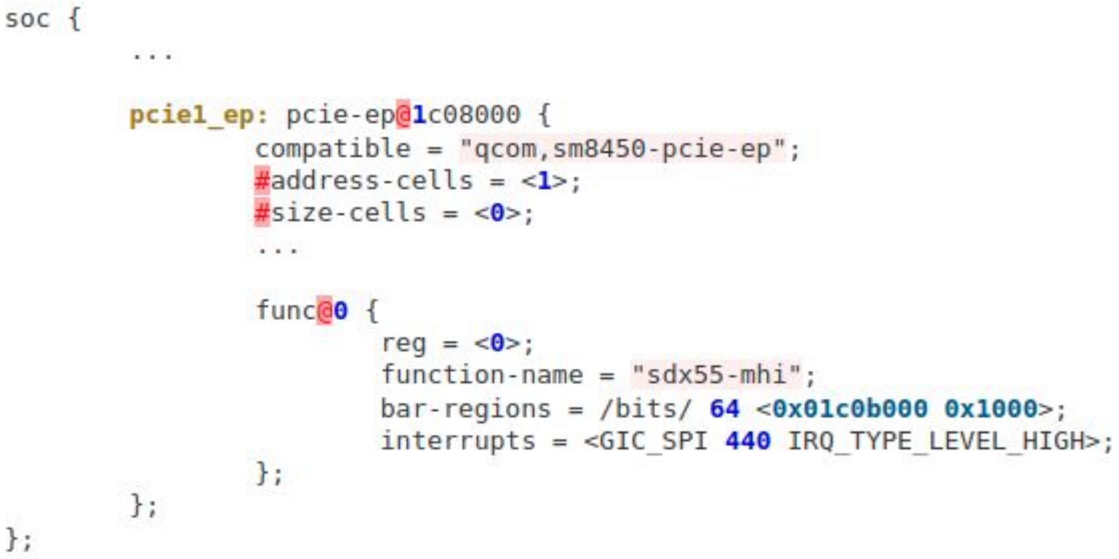
- Devicetree is currently not integrated with EPF drivers since most of the Endpoint functions are software implementations
- ConfigFS is currently used for specifying function configurations
- But there are Endpoint functions that has relevant hardware blocks. For example, MHI (Modem Host Interface) on Qualcomm chipsets
- MHI
  - MHI is a Qualcomm specific protocol that uses PCI as the physical layer for transferring data packets between PCI host and endpoint
  - MHI has a hardware implementation in Qualcomm chipsets supporting PCI Endpoint mode
  - Currently, PCI Endpoint Controller (EPC) devicetree node is used for fetching MHI specific data like BAR region and interrupt.

# Devicetree Integration

- **Proposal**
  - A devicetree node for MHI function
    - Child node of PCI Endpoint Controller (EPC) node
  - EPF device will be created for each function and later bound with EPF driver
  - Properties
    - reg
    - function-name
    - bar-regions
    - Interrupts
  - Due to the parent child relationship between EPC and EPF in devicetree, link can be established without ConfigFS intervention.

# Devicetree Integration

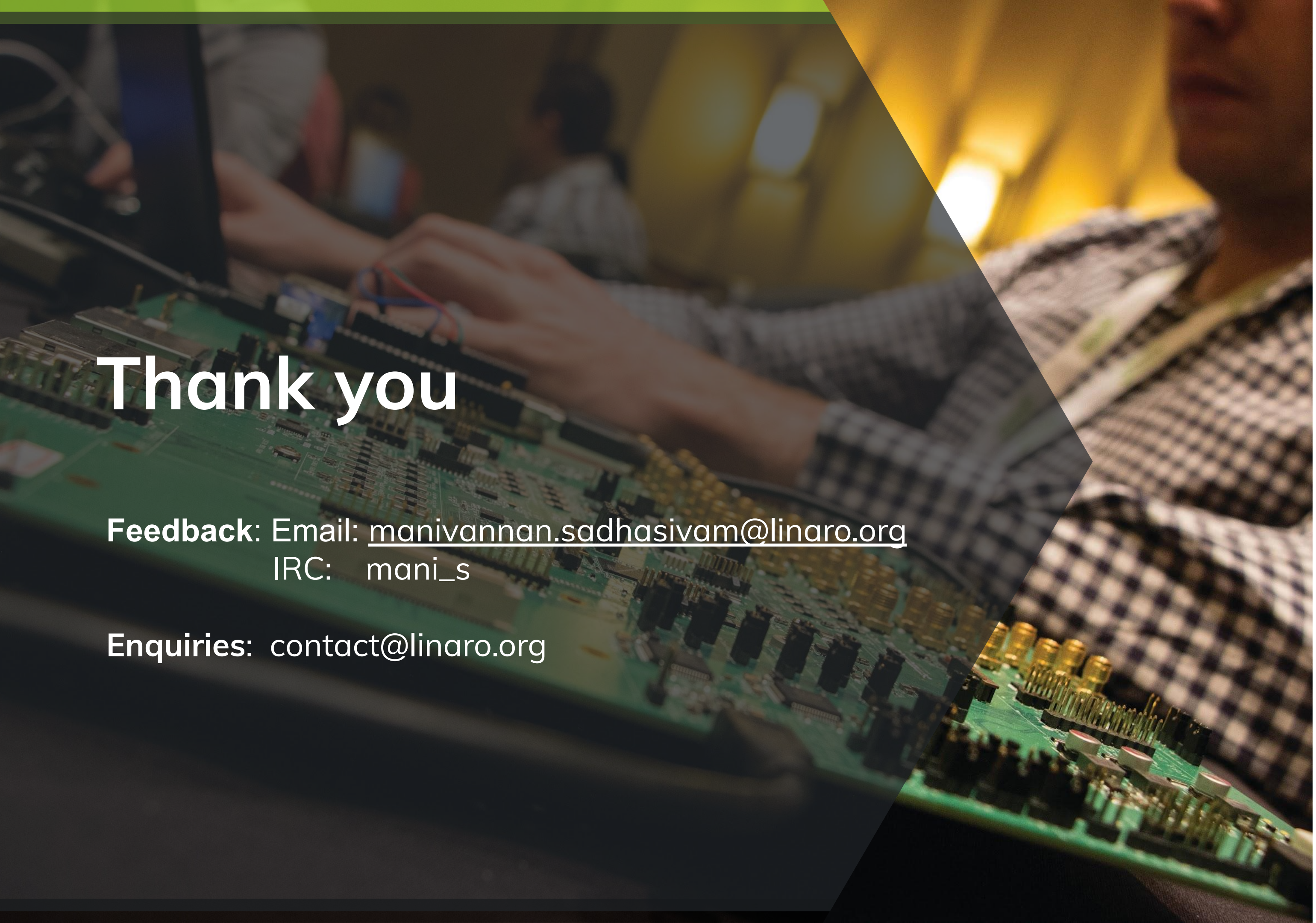
### Devicetree Binding



```
soc {
        ...

        pcie1_ep: pcie-ep@1c08000 {
                compatible = "qcom,sm8450-pcie-ep";
                #address-cells = <1>;
                #size-cells = <0>;
                ...

                func@0 {
                        reg = <0>;
                        function-name = "sdx55-mhi";
                        bar-regions = /bits/ 64 <0x01c0b000 0x1000>;
                        interrupts = <GIC_SPI 440 IRQ_TYPE_LEVEL_HIGH>;
                };
        };
};
```



# Thank you

**Feedback**: Email: manivannan.sadhasivam@linaro.org
IRC: mani_s

**Enquiries**: contact@linaro.org

Linaro
Developer Services