



On-Chip Interconnect API Proposal

Georgi Djakov / Vincent Guittot

LEADING
COLLABORATION
IN THE ARM
ECOSYSTEM

Linux Plumbers
Conference 2018
(PM & EA μ conf)

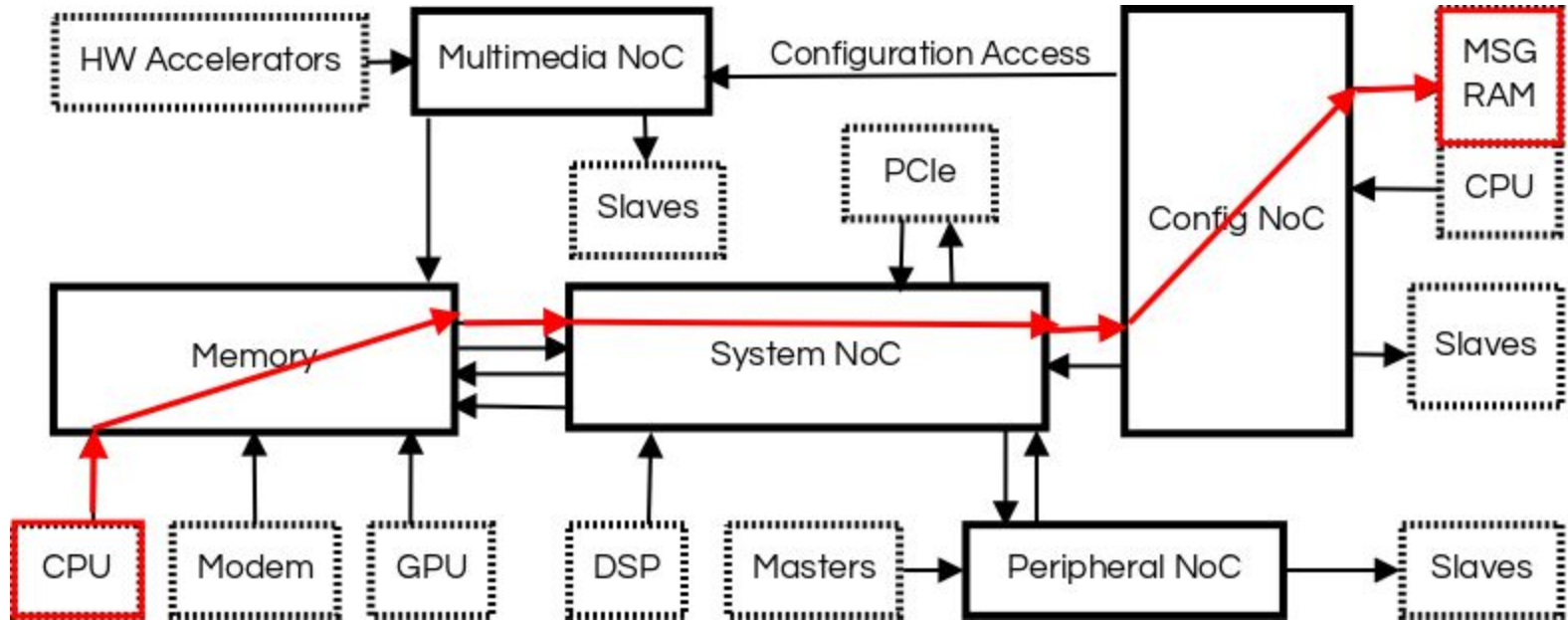
Agenda

- Some background
- Discussion
 - Tagging an interconnect path with properties
 - How a device can determine its own bandwidth needs

Some background

- Complexity of ARM SoCs
- The goal of the Interconnect API
 - A common solution to replace custom implementations by SoC vendors
 - Provider/consumer-based API - drivers express their bandwidth needs
 - Configure the system according to the demand and save power

An example topology



Tagging a path

- Tag a path with an additional property
- Active and sleep set example

How a device can determine its bandwidth needs

- The device can calculate the needed bandwidth
- Hardware specific value
- Predefined bandwidth values that scale with frequency
- Predefined bandwidth values according to use-case

Some devices can calculate their bandwidth needs

- UART (baud rate)
- I2C (Fast mode, High-speed mode, etc)
- SPI (clock rate)
- USB (USB specification version, data transfer type)
- SDHC (SD standard, clock rate, bus width)
- UFS (UFS standard version)
- Video Encoder/Decoder (video format, resolution, frame-rate)

Other devices use hard-coded internal values

- Hardware specific value (PRNG)
- Use-case specific value

Predefined bandwidth values - opp-bw DT property

Bandwidth and frequency are scaled together

```
opp_table0: opp_table0 {
    compatible = "operating-points-v2";

    opp-710000000 {
        /* Frequency = 710 MHz */
        op-hz = /bits/ 64 <710000000>;
        /* PORT0 average bandwidth = 0 KB/s, peak bandwidth = 7216000 KB/s */
        opp-bw-KBps-port0 = <0 7216000>;
        /* PORT1 average bandwidth = 0 KB/s, peak bandwidth = 1000000 KB/s */
        opp-bw-KBps-port1 = <0 1000000>;
    }
}
```

Predefined bandwidth values - use-case specific

Bandwidth and frequency are scaled separately

- GPU
 - Computationally intensive shaders require high GPU frequency but proportionally less bandwidth than one would expect
- CPU
 - Similar cases due to hit rates on caches
- Both CPU & GPU don't want their bandwidth requests to be done blindly based on core frequency

Thanks!