National Synchrotron Light Source II

Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

Centralized Deployment of EPICS Systems for Zynq-based Devices

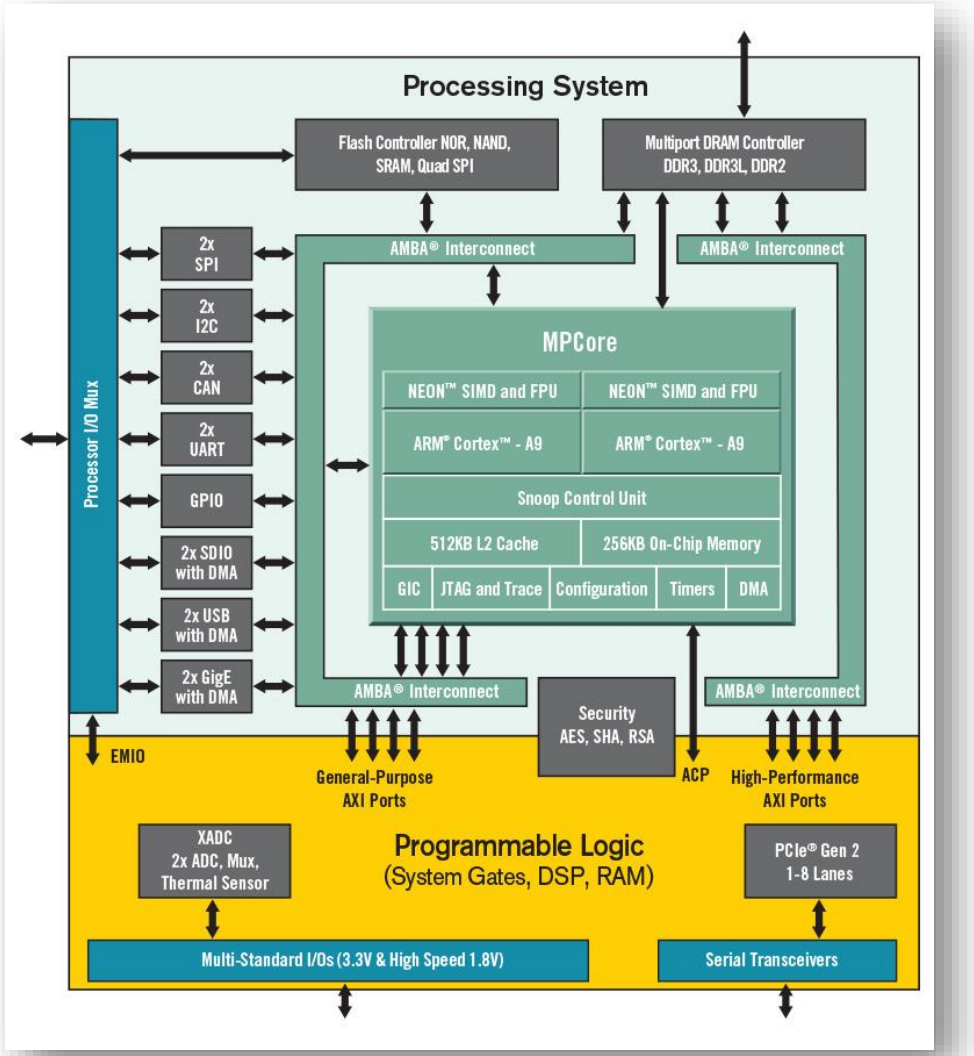
Ji Li, David Peter Siddons

EPICS Collaboration Meeting

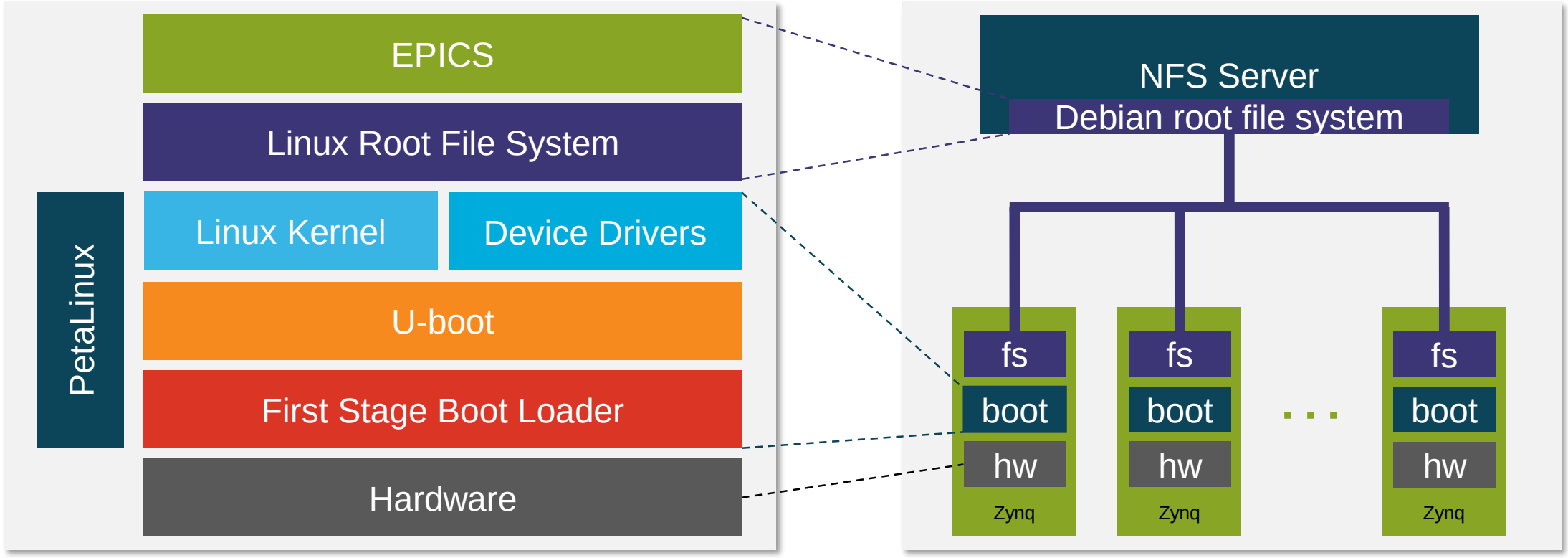
09/22/2022

Overview

- Zynq-based solutions become popular
  - Flexibility: **P**rocessing **S**ystem
  - Real-time performance and I/O versatileness: **P**rogramming **L**ogic
  - All-in-one solution
- Hundreds of Zynq devices deployed at NSLS-2
  - BPM, quadEM, Pizzabox, etc.
  - Problems:
    - Non-trivial build/maintenance work
      - Cross-compile/compile on ARM is time-consuming
    - Corrupted file system

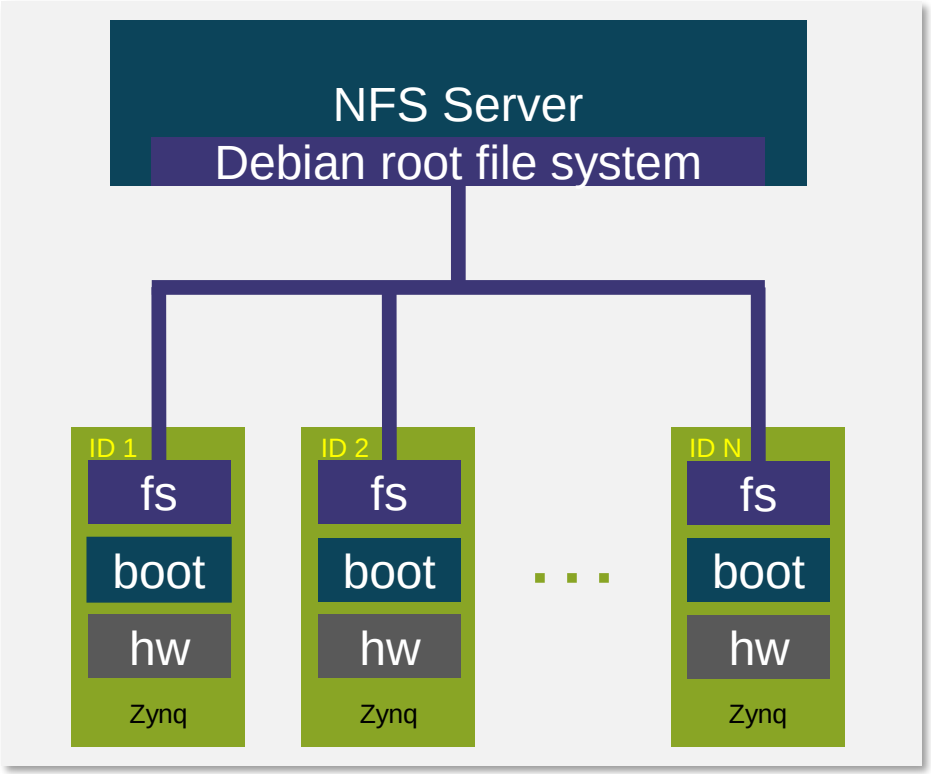


Zynq System


EPICS
Linux Root File System
Linux Kernel
Device Drivers
U-boot
First Stage Boot Loader
Hardware
PetaLinux
NFS Server
Debian root file system
fs
boot
hw
Zynq
fs
boot
hw
Zynq
fs
boot
hw
Zynq

Boot images

- Xilinx PetaLinux workflow
  - Configure to file system boot from NFS
  - `Boot.bin`, `image.ub`, `boot.scr`
  - Stored in SD Card
  - Build only once
- MAC addresses as Device IDs
  - Managed MAC address pool
  - Assign MAC address in U-boot boot script
  - Create a DHCP entry for each device



Root File System

- Standard Debian root file system
  - PetaLinux version →
  - Kernel version →
  - Debian version
- Multiple root file systems
  - PetaLinux versions
  - Architectures: 32-bit (Zynq-7000), 64-bit (Zynq UltraScale+)
- Export on NFS server
- Create only once

PetaLinux	2017.1	2019.1	2022.1
Linux	V4.9	V4.19	v5.15
Debian	9/Stretch	10 (Buster)	~11 (Bullseye) (v5.10)

Device Record

`/epics/data/zynq-dev.json`

- Information for all managed Zynq devices
  - Grouped by beamlines
  - DHCP validation
  - Name
    - Hostname/IOC name
  - PV name prefix
    - IOC start script, substitution files
- Updated for each new device


```
{
    "28ID1":
    [
        {
            "ip": "10.69.6.170",
            "mac": "0a:00:12:34:36:3f",
            "name": "xf28id1-quadem1",
            "sys": "XF:28ID1-ES",
            "dev": "{QuadEM:1}"
        },
        {
            "ip": "10.69.6.56",
            "mac": "0a:00:12:34:36:3f",
            "name": "xf28id1-ion1",
            "sys": "XF:28ID1-ES",
            "dev": "{Ion:1}"
        }
    ],

    "28ID2":
    [
        {
            "ip": "10.67.4.66",
            "mac": "0a:00:12:34:36:98",
            "name": "xf28id2-quadem1",
            "sys": "XF:28ID2-ES:1",
            "dev": "{QuadEM:1}"
        },
        {
            "ip": "10.67.4.76",
            "mac": "0a:00:12:34:52:3a",
            "name": "xf28id2-gem1",
            "sys": "28ID2-ES:2",
            "dev": "{Gem:1}"
        }
    ]
}
```

EPICS

- EPICS infrastructure

  `/epics/base-x.xx.x`

  `/epics/base-x.xx.x/support`

  - Multiple EPICS base versions
  - Multiple EPICS module versions
  - Build only once

- IOC Binary

  `/epics/modules/ion`

  - Build only once

- IOC Instance

  `/epics/iocs/xf28id1-ion1`

  - IOC name=hostname
  - Startup script and Substitution files
    - References to /`epics/modules/ion`
  - Autosave directory
  - `config`
  - Installed as services
    - `manage-iocs`
  - Create for each new device
  - No compilation needed

Start Up

- Select IOC at runtime - avoid writing to file system
  - Get name by IP address

```
name=`jq -c '.[]' zynq-det.json | awk '/'$my_ip'/ {print $0}' | jq -c
'.[]' | awk '/'$my_ip'/ {print $0}' | jq -r '.name'`
```

  - Start the IOC if the IOC is for the current device

```
if [ $name != $IOC ]; then exit fi
```

- Each device only start its own IOC
  - One instance

Summary

- Significantly simplifies deployment/maintenance
  - For each new device type
    - Build boot images, IOC binary
    - Create root file system (optional)
  - For each new device
    - Copy boot images, update boot script
    - Update device record, DHCP
    - Create IOC instance
    - Install IOC
    - Avoid compilation
- More robust file system.

Thank you