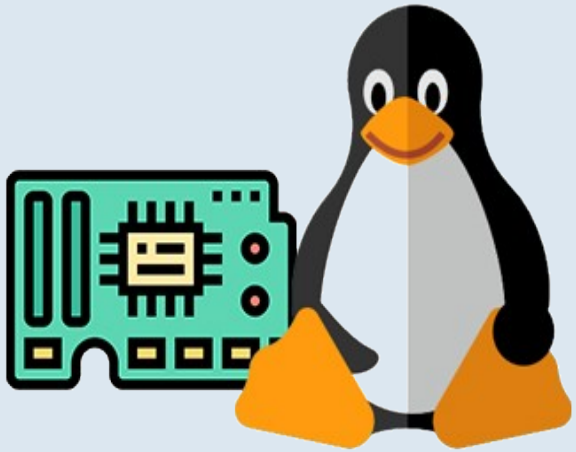
Part 4: Embedded Linux



Outline

- Bare metal
- BSP
- Linux
- Boot process

Without Linux

- Bare metal
  - No operating system
  - Write everything in Vitis yourself using a BSP ⟵ Hands-on 2
  - Difficult to access SD card
  - Ethernet…
  - Wifi?!?

Board Support Package (BSP)

- A collection of **libraries** and **drivers** that will form the lowest layer of your application **software stack**. Your software applications must link against or run on top of a given software platform using the **APIs** that it provides. Therefore, before you can create and use software applications in SDK (**Vitis**), you must create a board support package.

- **Standalone** - A simple, semi-hosted, and single-threaded environment that provides basic features such as standard input/output and access to processor hardware features
- **Xilkernel** - A simple and lightweight kernel that provides POSIX style services such as scheduling, threads, synchronization, message passing, and timers
- When a board support package is built, the following software components are automatically included into the library:
  - Standard C Libraries (libc.a, libm.a)
  - Device Drivers for all the peripherals in your design (libxil.a)

Linux

"Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones"
L. Torvalds



- Andrew Tanenbaum - 1987 – Minix (Mini-Unix) educational purposes
- Linus Torvalds – 1991
- **Open source**
- Runs on smartwatches, e-ink tablets, desktops, servers, supercomputers
- All run the **same kernel** code base
- “Everything is a **file**” [Unix]
- Git

Kernel – The heart of Linux

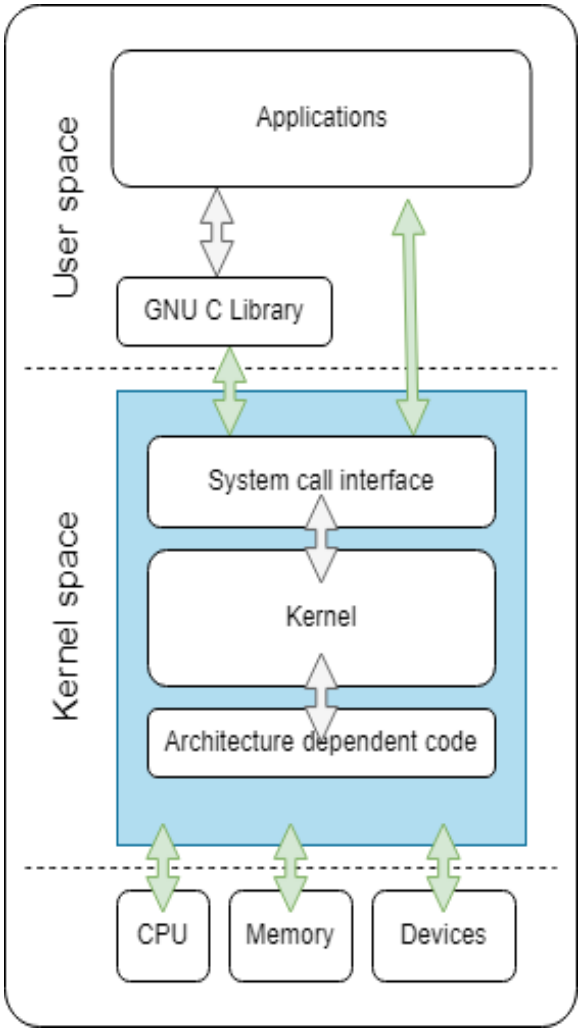
- The Kernel is the core interface between the computer's hardware and its processes

1. *Process management*: Schedules different applications to the CPU
2. *Memory management*: Protects the memory between applications
3. Handles *System calls* (fork, readfile, writefile, …)
4. *Device drivers*: interpreter between hardware and processes

- != OS (set of GNU Tools, gcc, …)

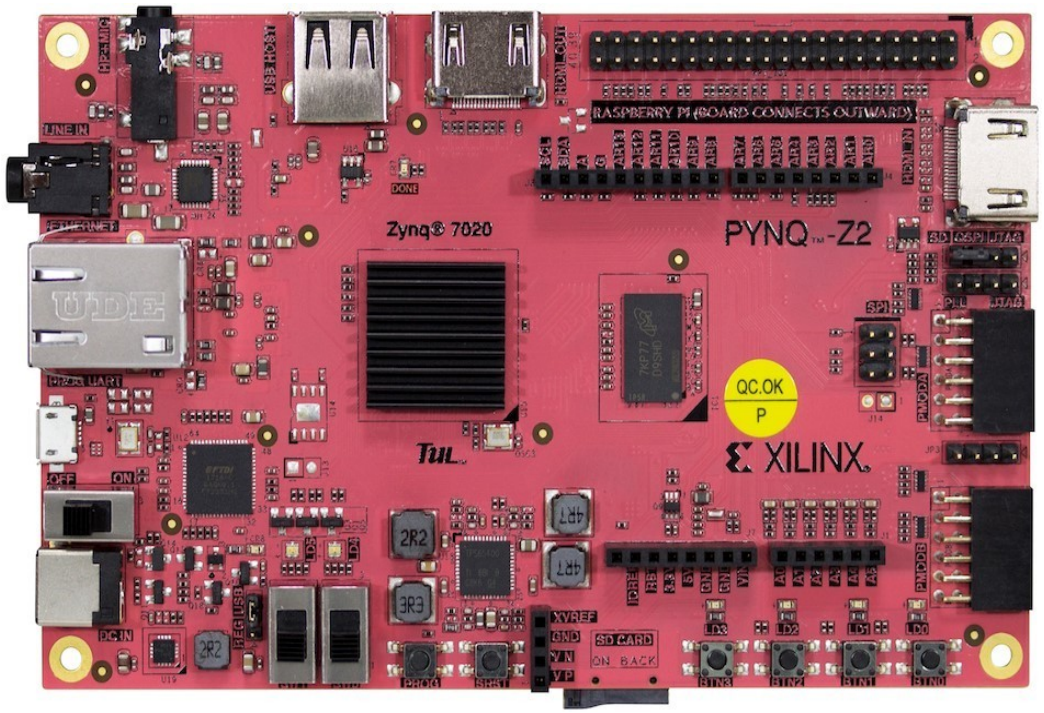
Kernel space vs User space

- Applications run in User space
- No direct access from User space to HW
- Device drivers run in kernel space
  - Reuse existing drivers eases development
- Memory Management Unit (MMU)
  - Each process uses memory from address zero upwards
  - MMU maps these “virtual addresses” to real physical memory
  - Simplifies development
  - Provides security: one process cannot access memory of another process
- Most RTOSes do not use this model (they have the same logical and physical addresses) Nucleus RTOS has an option for simplified MMU.



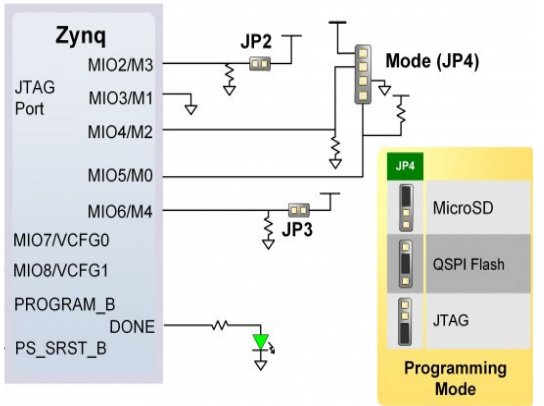
Boot process

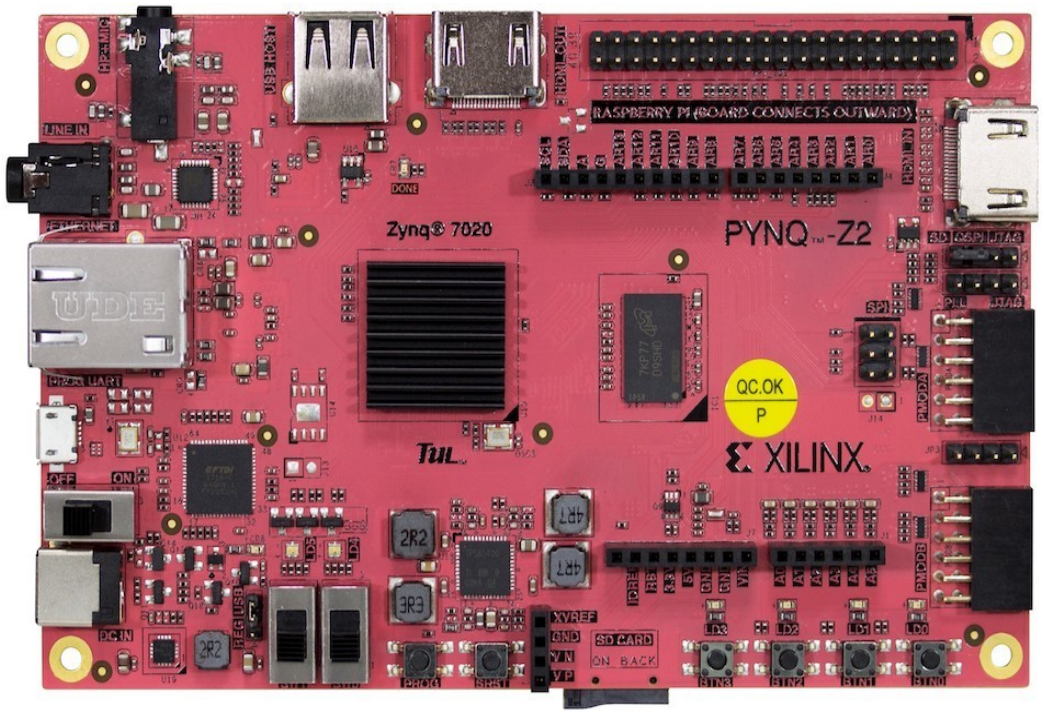
- Boot medium
- Boot stages of Linux
- Boot stages Bare metal



Boot process

- Boot medium
  - Jtag (USB cable)
  - Micro SD
  - Flash (16MB) - QSPI (can be programmed via Vivado SDK)
  - tftpboot

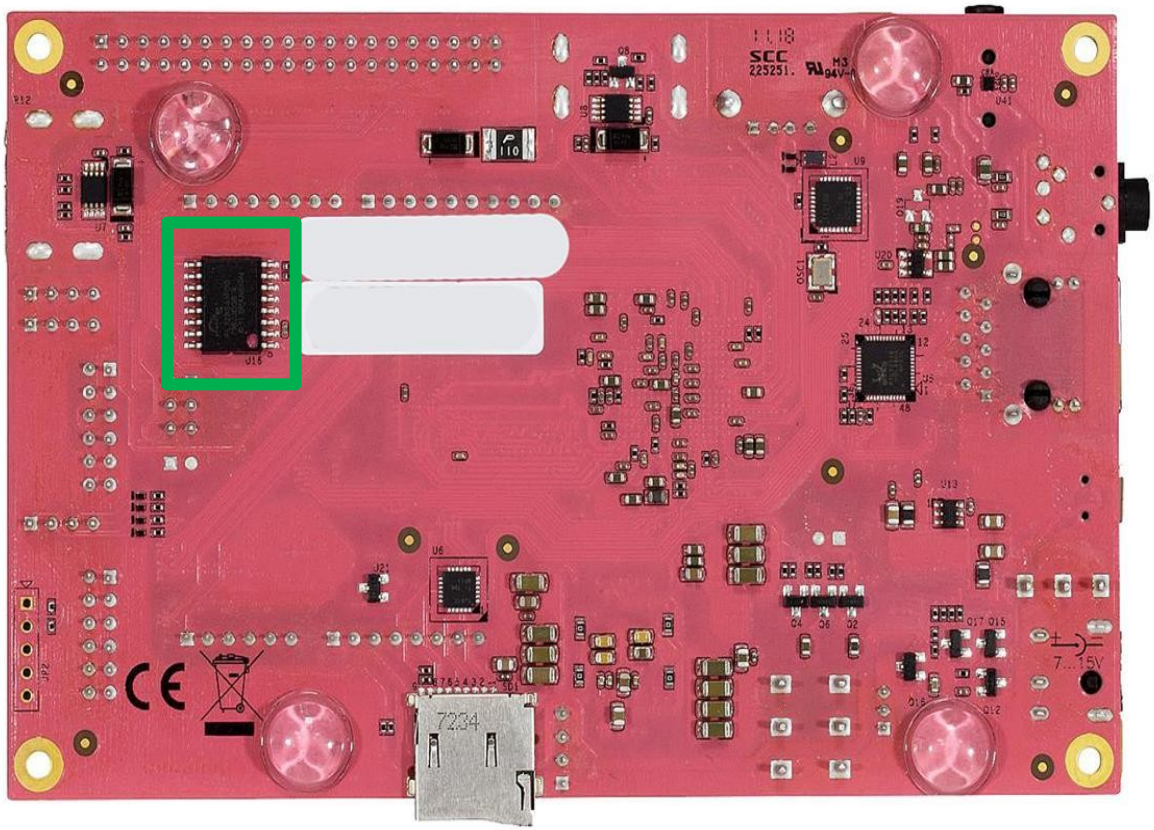




Boot stages of Linux – Stage 0

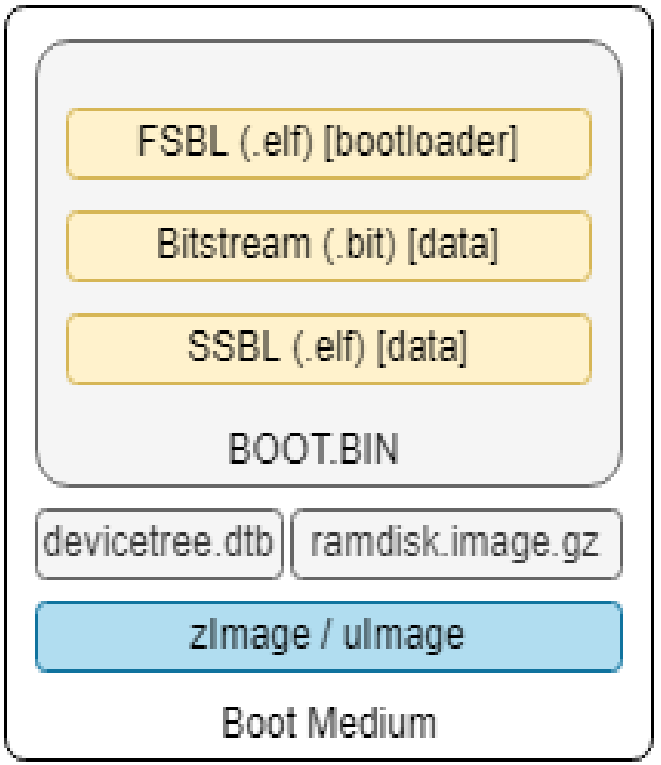
https://docs.xilinx.com/v/u/en-US/ug585-Zynq-7000-TRM (page 168, BootROM)

- Boot ROM (128KB)[S25FL128SAGMFI000]
- Power-On Self Test (POST)
- Check Boot medium
- Load First Stage Bootloader FSBL (512 bytes)



Boot stages of Linux – Stage 1

- Run First Stage Bootloader
  - Load bitstream and program FPGA
  - Load Second Stage Bootloader



.elf = Executable and Linkable Format

.bit = bitstream, generated by Vivado

BOOT.bin = generated by bootgen with boot.bif or by petalinux tools

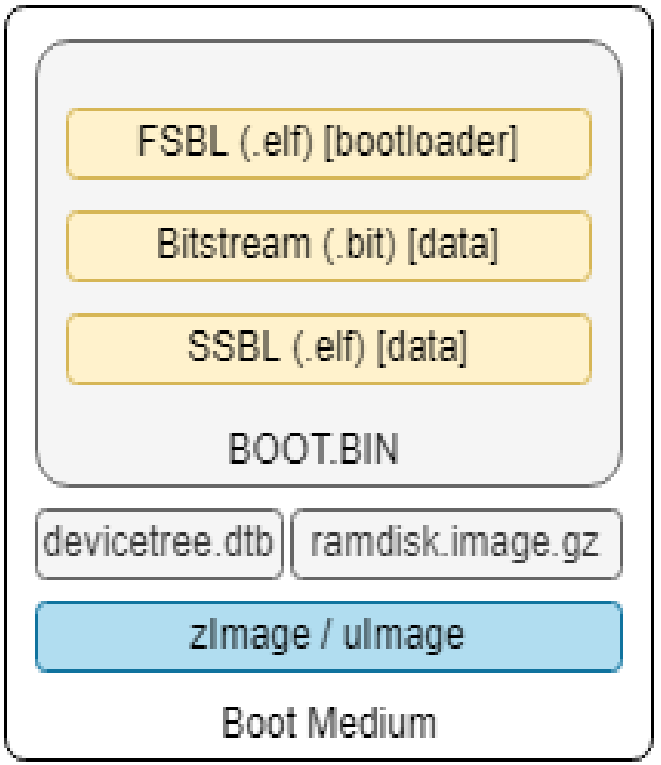
.dtb = devicetree binary, generated from .dts (devicetree source) with u-boot tools or by petalinux tools

zImage = compressed kernel image, uImage = uncompressed kernel image

mkimage = puts a wrapper around the image so U-Boot can handle it

Boot stages of Linux – Stage 2

- Run Second Stage Bootloader
  - Load Devicetree blob
  - Load Ramdisk (temporary root file system)
  - Load uImage (unzipped kernel image)

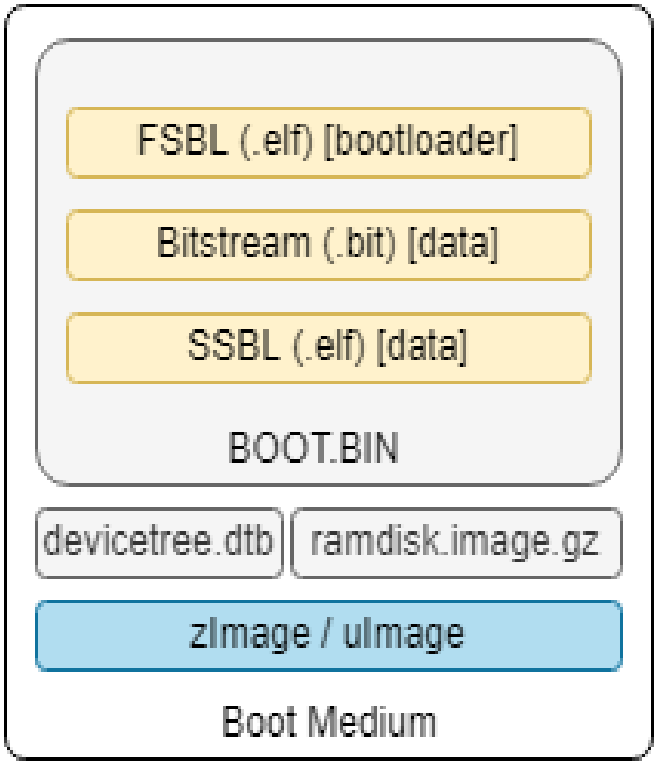


The devicetree is a data structure and language for describing hardware, More specifically, it is a description of hardware that is readable by an operating system so that the operating system doesn't need to hard code details of the machine.

=> The kernel no longer contains the description of the hardware.

Boot stages of Linux – Stage 3

- Init process /sbin/init (6 runlevels)
- Login prompt
- Startup scripts



- Runlevels :
  0 – halt
  1 – single user text mode
  2 – not used
  3 – multi user text mode
  4 – not used
  5 – multi user graphics mode
  6 – reboot

Boot stages – Bare metal

- Stage 0 – same
- Stage 1 – same
- SSBL is the program you created in Vitis
- No stage 3

