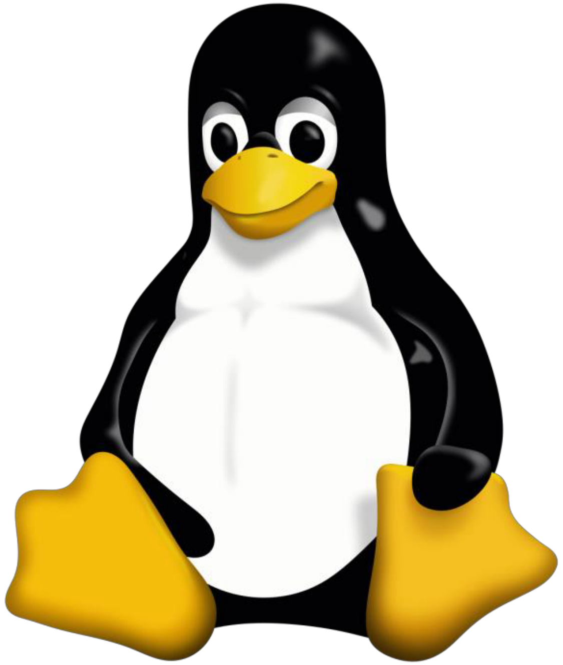


Linux Device Driver

Sunbeam Infotech

OS Booting

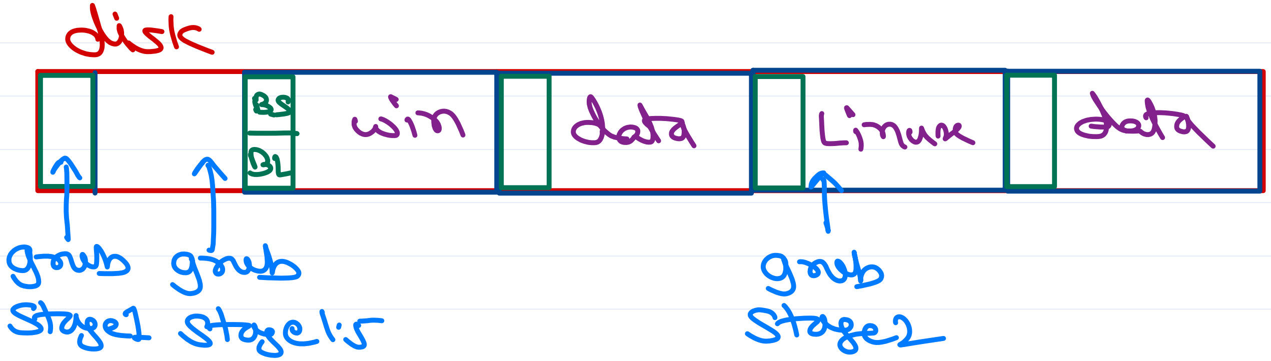
OS Booting

1. POST
2. Bootstrap Loader
3. Device → Bootloader
4. User → select OS
5. BL → Bootstrap program
6. OS kernel

Windows Booting

1. POST
2. Bootstrap Loader BCD
3. Device → Bootloader -bootmgr
4. User → select OS
5. BL → Bootstrap program → winload.exe
6. OS kernel → ntoskrnl.exe

Linux Booting



① ROM → POST

② ROM → Bootstrap Loader

③ Device → GrUB stage 1

④ GrUB Stage 1.5
(with FS driver).

⑤ GrUB Stage 2
↳ show options from
grub.cfg

⑥ user selection
↳ linux kernel +
↳ initrd

⑦ Kernel → Self extract
and use initrd as initial FS.
↳ initrd contains drivers for disk and other chipset (which are not statically compiled in kernel, but needed for booting).
↳ kernel read disk and attach to on disk FS.

⑧ Further kernel processes execute.
↳ 1st user space process - init/systemd (pid=1).

⑨ Start services as per runlevels/targets.

Linux Booting

Linux process 1

→ init (older kernels)

→ systemd (modern kernels)

init

✓ designed for uni-processor

✓ services (daemons) are stared one after another.

✓ booting slower

✓ initialization scripts (/etc/init)

* Booting run levels

① single user (root)

② multi user (login security)

③ networking

④ ~~reserved~~

⑤ graphics (gui)

```
> init 0 (shutdown)
> init 6 (reboot)
```

Systemd

✓ designed for multi-processor.

✓ services (daemons) are started in parallel.

✓ faster booting

✓ initialization scripts (units).

* Booting targets.

① rescue.target (single user mode)

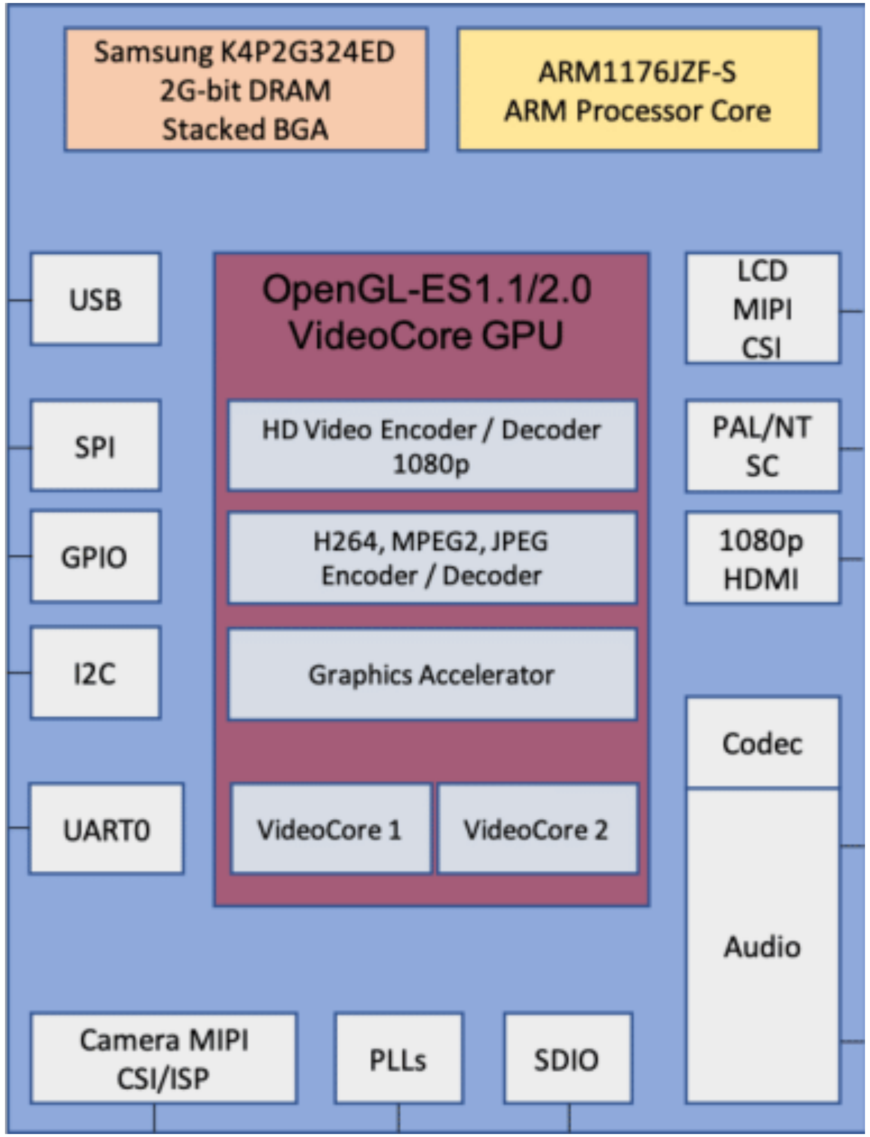
② multi-user.target (login security + networking)

③ graphical target.

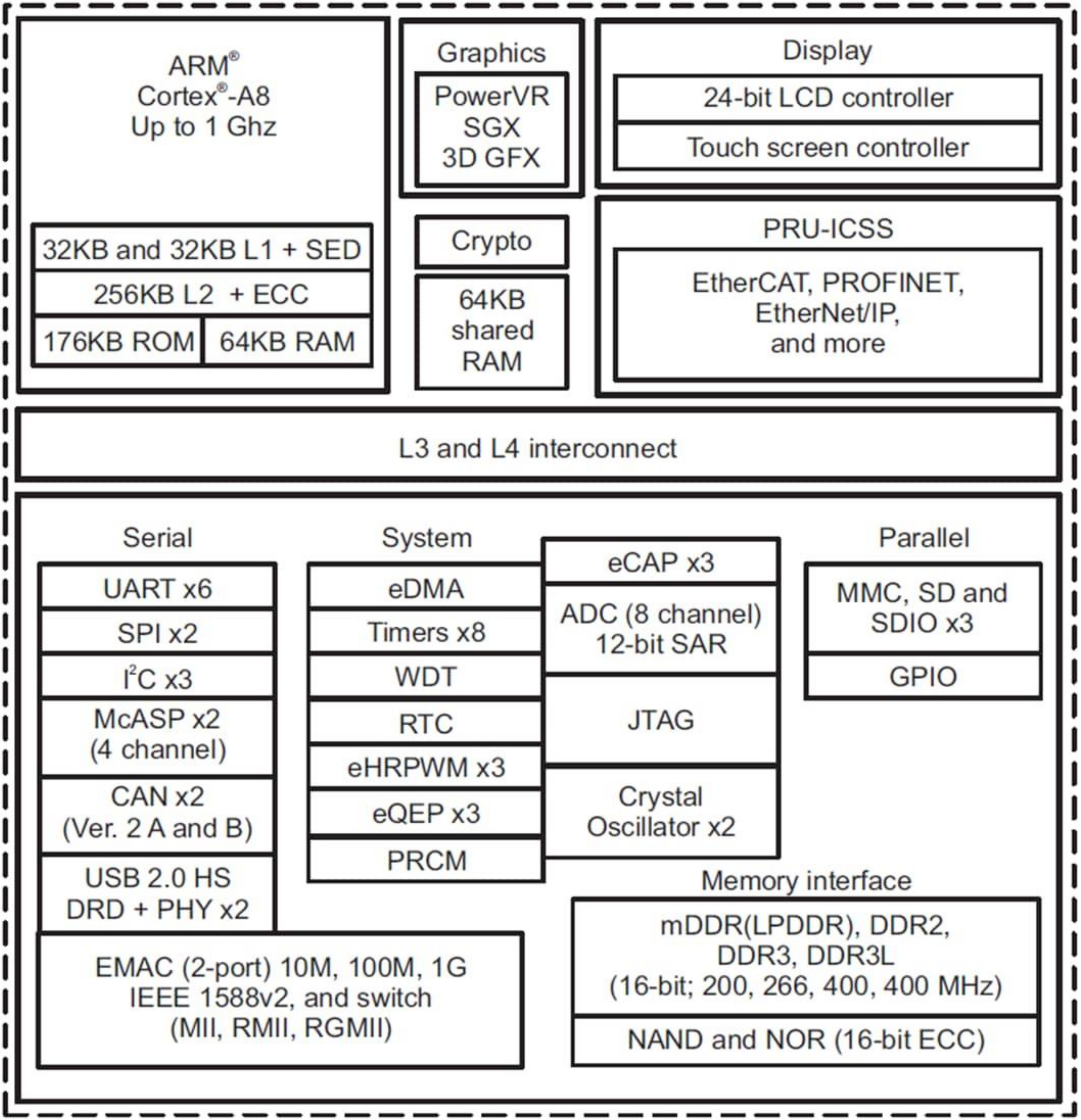
```
> man 7 bootup
```

BCM2835 & AM335x

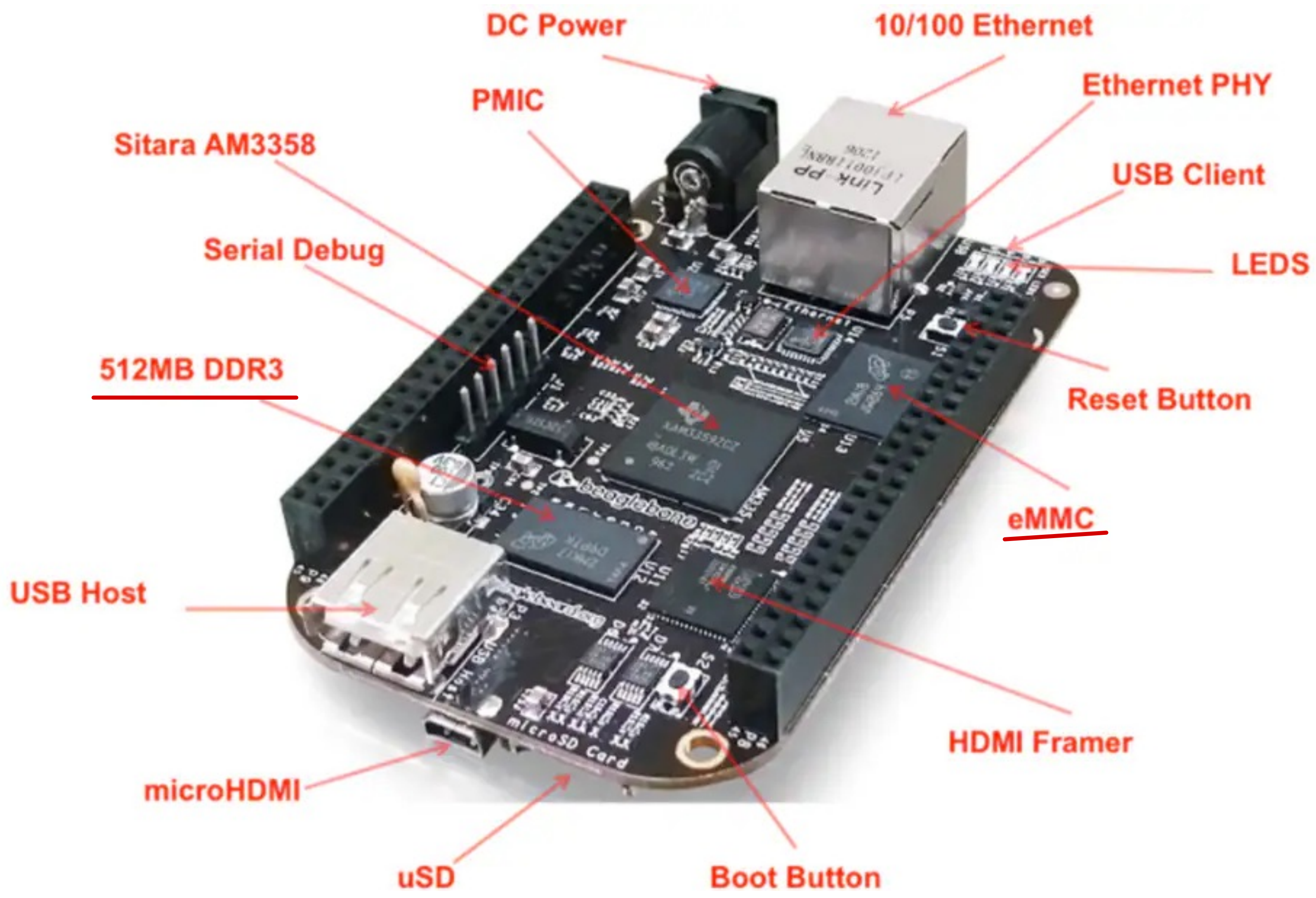
RPi



BBB →



BeagleBone Black



Cortex-A8

ROM - 176 KB

RAM - 64 KB
+ 64 KB

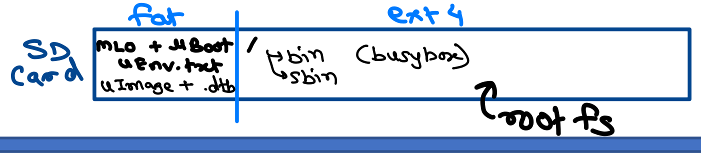
External RAM - DDR3
- 512 MB

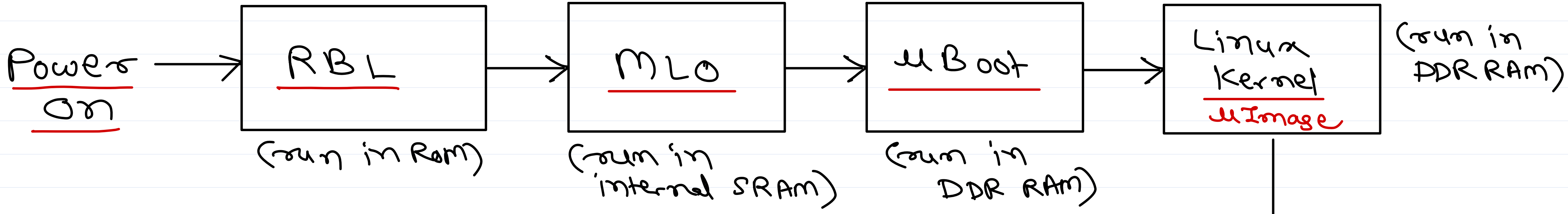
eMMC - 4GB card.

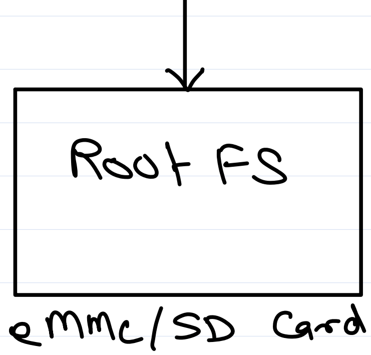
Extenal MMC card
- 8 GB / 16 GB...

BeagleBone Linux Booting

AM335x (ARM SoC)







① RBL

- ✓ ROM Boot Loader
- ✓ Given by manufacturer and hard-coded in flash.
- ✓ Does following
  - Default clock init
  - Stack setup (for all modes)
  - Set up WDT (3 mins)
  - PLL config for clocks
  - decide boot device (emmc/SD)
  - Get SPL from mmc and invoke it.

BBB board : XTAL = 24 MHz

② SPL/MLO

- ✓ Secondary Program Loader (Memory Loader)
- ✓ Primary or First stage Bootloader.
- ✓ Run in uC SRAM.
- ✓ Part of UBoot loader
- ✓ DDR RAM initialization
- ✓ Get UBoot from mmc & invoke it

③ UBoot

- ✓ Run in DDR RAM
- ✓ Get config from uEnv.txt
- ✓ Invokes Linux Kernel & FDT

④ Linux Kernel

- ✓ Load in DDR RAM & extract itself.
- ✓ Loads RootFileSystem from mmc card.

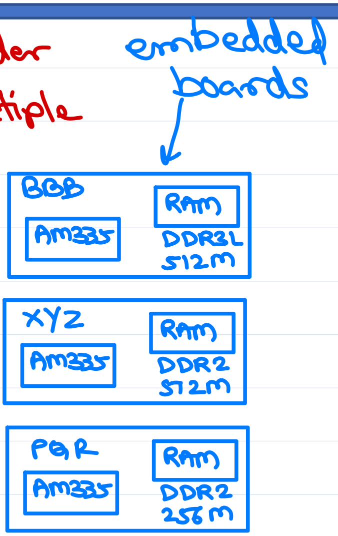
⑤ Root FS

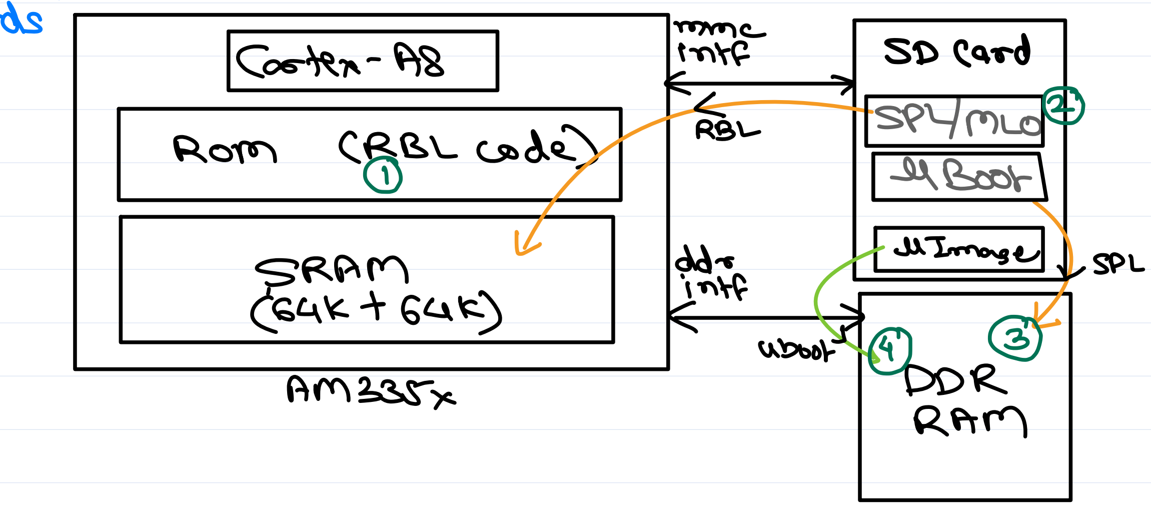
- ✓ Contains all system binaries.

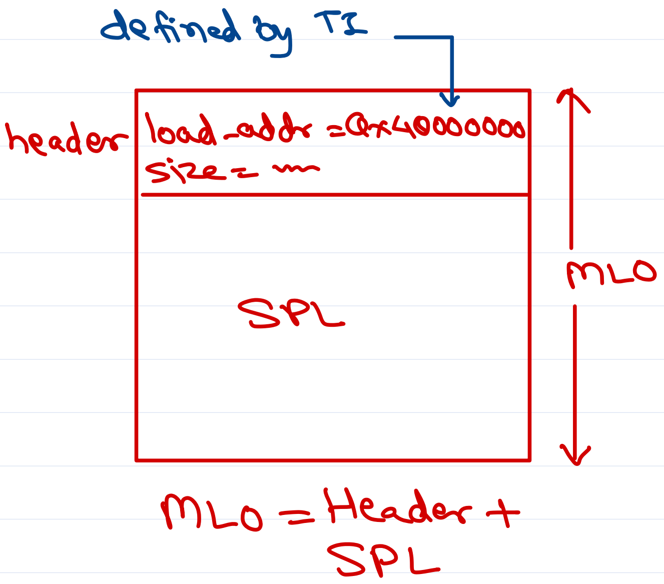
BeagleBone Linux Booting

AM335x ROM Bootloader can load from multiple devices.

1. emmc
2. SDCard
3. SPI
4. UART
5. Ethernet
6. USB
7. NAND Flash







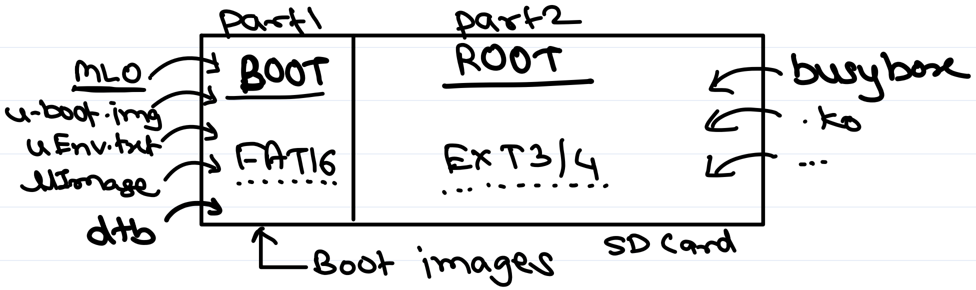
Boot seq is defined as by sysboot [4:0] - TRM.

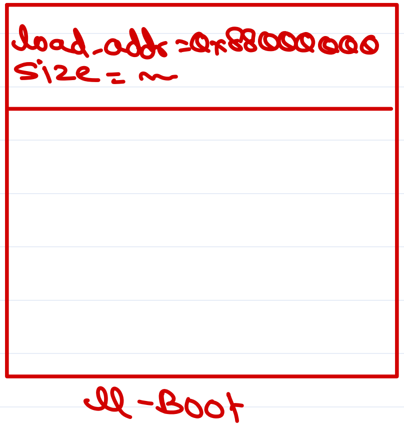
On BBB boot seq can be changed using S2 switch.

(A) S2 released [11100]	(B) S2 pressed [11000]
1. mmc1 (emmc)	1. SPI0
2. mmc0 (SD)	2. mmc0 (SD)
3. UART0	3. USB0
4. USB0	4. UART0

Q. why MLO is loaded into SRAM & uBoot in DDR RAM?

Q. what is need of SPL? why RBL do not load uBoot directly?

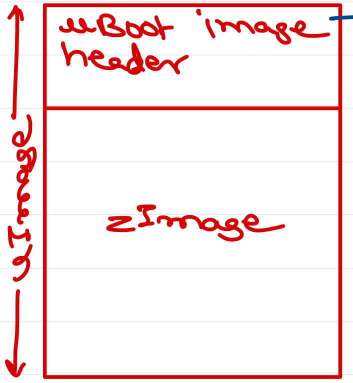




BeagleBone Linux Booting

mkimage

uImage = zImage + uBoot Image header(64 bytes)



- ① magic
- ② crc
- ③ time(creation)
- ④ size
- ⑤ load addr
- ⑥ entry point
- ⑦ arch
- ⑧ name
- ⑨ compression
- ⋮

linux kernel image (in rfs)

xxd -l 64 uImage

uBoot code

basic device/peri initialization

load the kernel & devicetree

boot_jump_linux

↓

linux kernel source code

decompress kernel

arch specific initialization

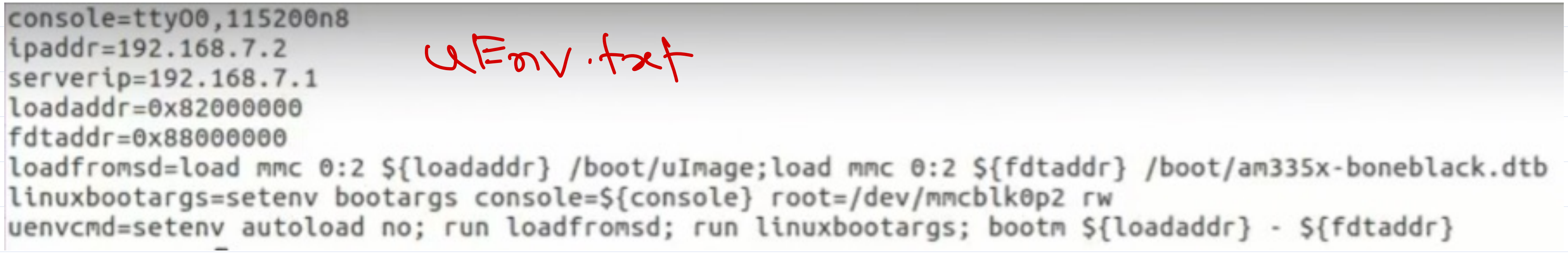
- ✓ lookup arch & invoke arch mm
- ✓ load device tree & init peri

main.c - start_kernel()

- ✓ kernel data struct & subsystems initialization
- ✓ rest_init() → kernel threads

① init ② kthreadd

uEnv.txt



```
console=ttyO0,115200n8
ipaddr=192.168.7.2
serverip=192.168.7.1
loadaddr=0x82000000
fdtaddr=0x88000000
loadfromsd=load mmc 0:2 ${loadaddr} /boot/uImage;load mmc 0:2 ${fdtaddr} /boot/am335x-boneblack.dtb
linuxbootargs=setenv bootargs console=${console} root=/dev/mmcblk0p2 rw
uenvcmd=setenv autoload no; run loadfromsd; run linuxbootargs; bootm ${loadaddr} - ${fdtaddr}
```

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>