

# MXS without Freescale tools

Marek Vašut <marex@denx.de>

October 24, 2013

- ▶ General-purpose industrial-grade CPU
- ▶ 454MHz ARMv5 core
- ▶ Based on Sigmatel design  
STMP3780 → i.MX233 → i.MX28 → i.MX6
- ▶ Plenty of useful peripherals (some re-used on i.MX6)

- ▶ General-purpose industrial-grade CPU
- ▶ 454MHz ARMv5 core
- ▶ Based on Sigmatel design  
STMP3780 → i.MX233 → i.MX28 → i.MX6
- ▶ Plenty of useful peripherals (some re-used on i.MX6)
- ▶ **Poor vendor software support**

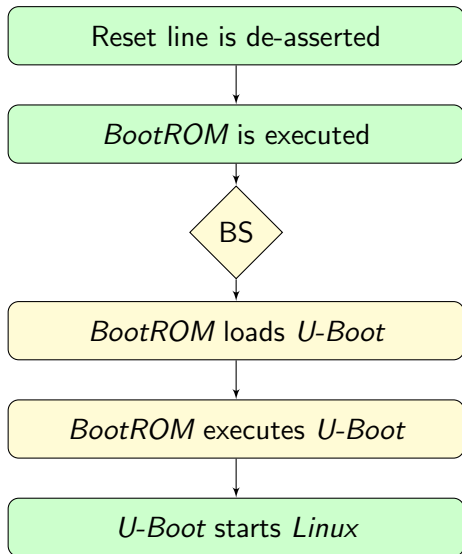
# Creating *U-Boot* image the old way

- ▶ Download *imx-bootlets* and *elftosb*
- ▶ Download *U-Boot*
- ▶ Compile *elftosb* (doesn't work on AMD64)
- ▶ Hack *imx-bootlets* for your board
- ▶ Compile *imx-bootlets* (you need lucky toolchain)
- ▶ Compile *U-Boot*
- ▶ Assemble!
- ▶ `elftosb -zf imx28 -c bootlets.bd -o u-boot.sb`
- ▶ Produce a bootable image: (The command is unknown)
- ▶ Load this onto your i.MX28 system (mfgtool?)

## Creating *U-Boot* image the old way

- ▶ Download *imx-bootlets* and *elftosb*
- ▶ Download *U-Boot*
- ▶ Compile *elftosb* (doesn't work on AMD64)
- ▶ Hack *imx-bootlets* for your board
- ▶ Compile *imx-bootlets* (you need lucky toolchain)
- ▶ Compile *U-Boot*
- ▶ Assemble!
- ▶ `elftosb -zf imx28 -c bootlets.bd -o u-boot.sb`
- ▶ Produce a bootable image: (The command is unknown)
- ▶ Load this onto your i.MX28 system (mfgtool?)
- ▶ This looks like **poor job**

# Booting the i.MX28



- ▶ Replacing *imx-bootlets* by *U-Boot*
- ▶ Replacing *elftosb* by *U-Boot*
- ▶ Replacing *mfgtool* by MXSLDR

They are responsible for:

- ▶ Initializing the POWER block
- ▶ Initializing the DRAM controller
- ▶ Booting Linux

Everyone has their own "version"



## *U-Boot* version:

- ▶ Discarded bootlets early during the i.MX28 *U-Boot* port
- ▶ Re-implemented the *imx-bootlets* code
- ▶ There is now one version of code that covers all boards
- ▶ The code is clean, documented and understandable
- ▶ See `arch/arm/cpu/arm926ejs/mxs/spl.*`
- ▶ It uses the SPL framework!

Interesting bits:

- ▶ We can jump back into the BootROM
- ▶ The BootROM handles the loading from any media for us
- ▶ One BootStream image fits all supported media

- ▶ We managed to obtain partial SB-format specification
- ▶ We initially implemented standalone tool *MXSSB*
- ▶ *MXSSB* was merged into mainline *U-Boot*
- ▶ *MXSSB* has 10 times less lines of code than *elftosb*
- ▶ Written in pure C , depends only on OpenSSL
- ▶ Well integrated into *U-Boot* build process
- ▶ Does not support encrypted/signed boot (yet)

We still need to wrap the image more for NAND and SD boot

- ▶ The mxsboot tool in *U-Boot* does that for us
- ▶ The mxsboot tool generates NAND/SD/SPI images

*U-Boot* build process will produce the bootable image for i.MX28

- ▶ Without external dependencies (but OpenSSL)
- ▶ All well integrated in one place
- ▶ All happens with two commands:

```
make board u-boot.sb
```

```
./tools/mxsboot sd u-boot.sb u-boot.SD
```

The MX23/MX28 BootROM supports USB firmware upload

- ▶ The `mxsldr` tool can upload `u-boot.sb`
- ▶ Depends only on `libusb`
- ▶ Connect the board via USB OTG port to PC and use `mxsldr`

Thank you for your attention.

Questions?