U-Boot startup sequence

Status, Thoughts, Discussion

U-Boot Summit ELCE 2015
Simon Glass, sjg@chromium.org
Agenda

● Status quo
● Recent changes
● Grand plans
● Discussion
Status quo (U-Boot proper on ARM)

- start
- CPU
- stack
- board_init_f_mem (gd, early malloc)
- board_init_f
- dram_init
- Banner
- Reserve memory
- relocate_code
- board_init_r
- main_loop
Status quo (SPL on ARM)

start

CPU

stack setup

board_init_f_mem (gd, early malloc)

board_init_f

dram_init

load U-Boot

jump

stack in DRAM

board_init_r

spl_board_init
Recent changes

- **SPL**
  - board_init_f_mem()
  - Common board_init_f()
  - Do as little as possible before board_init_f() - i.e. no DRAM init
  - Return normally, instead of jumping to board_init_r()
  - CONFIG_SPL_STACK_R to relocate stack when DRAM is ready

- **U-Boot proper**
  - crt0.s common start-up code
  - Driver model taking over / dropping on some init sequence calls
Grand plans

- **SPL**
  - Move all boards to use a common `board_init_f()`
    - Put board-specific init in `board_early_init_f()`, or similar
  - Smaller `printf()` by default

- **U-Boot proper**
  - Start to deprecate more of the post-relocation init sequence
  - `probecpu()`, `timer_intt()`, `serial_init()`, `initr_spi()`, `initr_mmc()`, etc.
  - Generic ELF relocation

- **Crazy Stuff**
  - Common init for SPL and U-Boot proper
Discussion