Kbuild & Kconfig for U-Boot

Masahiro Yamada

Panasonic Corporation

October 13, 2014

Linux Kernel's build system

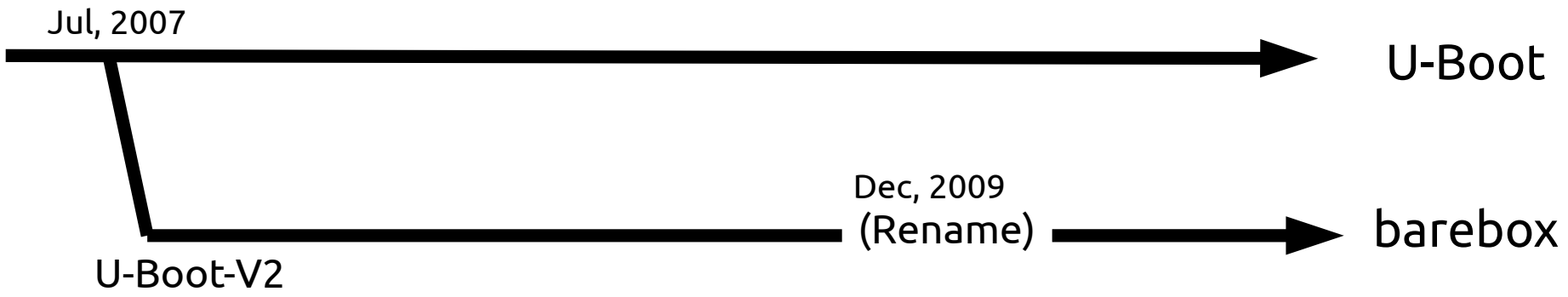
- Kbuild
  - Simple Makefiles
  - Readable log
  - Precise dependency tracking
- Kconfig
  - Easy to change/browse configuration
  - Clear dependency between CONFIGs
  - Help docs in Kconfig rather than a README

I am not the first man

Several suggestions, patches, repositories before me

- Holger Schurig: “Linux Kernel Config 1.2 for u-boot?”
  Nov, 2002 http://thread.gmane.org/gmane.comp.boot-loaders.u-boot/7132/focus=7139
- Carsten Schlote: “[kconfig] Second draft available for download”
  Jun, 2007 http://thread.gmane.org/gmane.comp.boot-loaders.u-boot/29516
- Saschar Hauer: “U-Boot-V2”
  Jun, 2007 http://thread.gmane.org/gmane.comp.boot-loaders.u-boot/29597
- Grant Likely: “[RFC] u-boot migration to kconfig”
  Sep, 2007 http://thread.gmane.org/gmane.comp.boot-loaders.u-boot/31766
- Simon Glass: “RFC: Add Kbuild system to U-Boot”
  May, 2014 http://thread.gmane.org/gmane.comp.boot-loaders.u-boot/160983/focus=162084

U-Boot V2 (barebox) by Saschar Hauer



Interesting features (mostly inspired from Linux):

- A POSIX based file API
- Shell commands like ls/cd/mkdir/echo/cat,...
- Real filesystem
- Kbuild makefiles and Kconfig configuration system
- Driver model
- User-mode U-boot (sandbox)
- Editor
- Some other goodies

But...

- Existing board supports had to be ported to the new code base

How to port?

- Do not break any features and boards
- Keep up with the mainline

How ?

Do not try to do everything at once!

Let's start with what we can do!

Long way to Promised Land

Too big change

Full Support of Kbuild & Kconfig

Step1

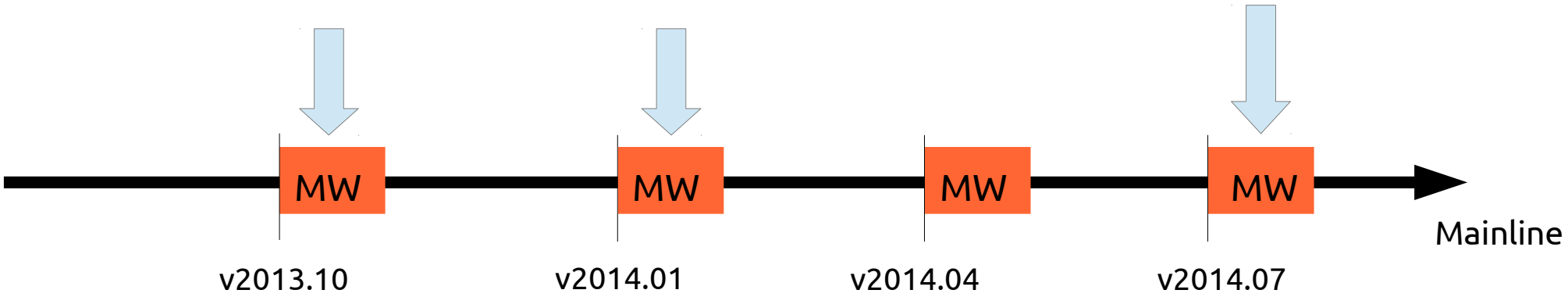
Kbuild sytle Makefile

Step 2

Kbuild

Step 3

Kconfig



Step 1: Kbuild Style Makefile (Nov. 2013)

Before

```
include $(TOPDIR)/config.mk

LIB  =  $(obj)lib$(SOC).o

SOBJS     := lowlevel_init.o

COBJS     += board.o
COBJS     += clock.o
COBJS     += mem.o
COBJS     += sys_info.o
ifdef CONFIG_SPL_BUILD
COBJS-$(CONFIG_SPL_OMAP3_ID_NAND)  += spl_id_nand.o
endif

COBJS-$(CONFIG_DRIVER_TI_EMAC)     += emac.o
COBJS-$(CONFIG_EMIF4)    += emif4.o
COBJS-$(CONFIG_SDRC)     += sdrc.o
COBJS-$(CONFIG_USB_MUSB_AM35X)     += am35x_musb.o

SRCS := $(SOBJS:.o=.S) $(COBJS:.o=.c)
OBJS := $(addprefix $(obj),$(COBJS) $(COBJS-y) $(SOBJS))

all:  $(obj).depend $(LIB)

$(LIB):   $(OBJS)
     $(call cmd_link_o_target, $(OBJS))

#########################################################################

# defines $(obj).depend target
include $(SRCTREE)/rules.mk

sinclude $(obj).depend

#########################################################################
```

Common for all Makefiles

Much simpler Makefile!

- Move the common parts to scripts/Makefile.build
- Rename `COBJS, SOBJS => obj-y`

After

```
obj-y  := lowlevel_init.o

obj-y  += board.o
obj-y  += clock.o
obj-y  += mem.o
obj-y  += sys_info.o
ifdef CONFIG_SPL_BUILD
obj-$(CONFIG_SPL_OMAP3_ID_NAND)  += spl_id_nand.o
endif

obj-$(CONFIG_DRIVER_TI_EMAC)  += emac.o
obj-$(CONFIG_EMIF4) += emif4.o
obj-$(CONFIG_SDRC)  += sdrc.o
obj-$(CONFIG_USB_MUSB_AM35X)  += am35x_musb.o
```

[SIDENOTE] How to confirm the correctness?

Compare MD5SUM

- include/timestamp.h

  Set constant strings to

  - `U_BOOT_DATE`
  - `U_BOOT_TIME`

```
 #ifndef DO_DEPS_ONLY
-#include "generated/timestamp_autogenerated.h"
+#define U_BOOT_DATE "DUMMY"
+#define U_BOOT_TIME "DUMMY"
 #endif
```

- include/version.h

  Set constant strings to

  - `PLAIN_VERSION`
  - `U_BOOT_VERSION`
  - `CC_VERSION_STRING`
  - `LD_VERSION_STRING`

```
 #ifndef DO_DEPS_ONLY
-#include "generated/version_autogenerated.h"
+#define PLAIN_VERSION "DUMMY"
+#define U_BOOT_VERSION "DUMMY"
+#define CC_VERSION_STRING "DUMMY"
+#define LD_VERSION_STRING "DUMMY"
 #endif
```

- MAKEALL

  Display MD5SUM of

  - `./u-boot.bin`
  - `spl/u-boot-spl.bin`
  - `tpl/u-boot-tpl.bin`

Step2 Kbuild (Feb, 2014)

- Import core scripts from Linux
- Adjust for SPL/TPL support

Output directory of objects

Normal: `./*`

SPL: `spl/*`

TPL: `tpl/*`

(scripts/Makefile.build was adjusted)

Step3 Kconfig (Aug., 2014)

- Import Kconfig (scripts/kconfig/*)
- Adjust for SPL/TPL support
- Adjust for coexistence of config headers

  The configuration system should support:

  - CONFIG_* in header files (include/configs/<board>.h)
  - CONFIG_* in Kconfig

Pre-kconfig Configuration System

Before Kconfig

(1) `$ make <board>_config`

boards.cfg —mkconfig→

- include/config.mk
- include/config.h
- arch/${ARCH}/include/asm/arch

  => arch/${ARCH}/include/asm/arch-${SOC}

(2) `$ make`

include/common.h

(include/configs/<board>.h)

—tools/scripts/define2mk.sed→

- include/autoconf.mk
- include/autoconf.mk.dep
- include/spl-autoconf.mk
- include/tpl-autoconf.mk

Configuration by Kconfig

After

(1)

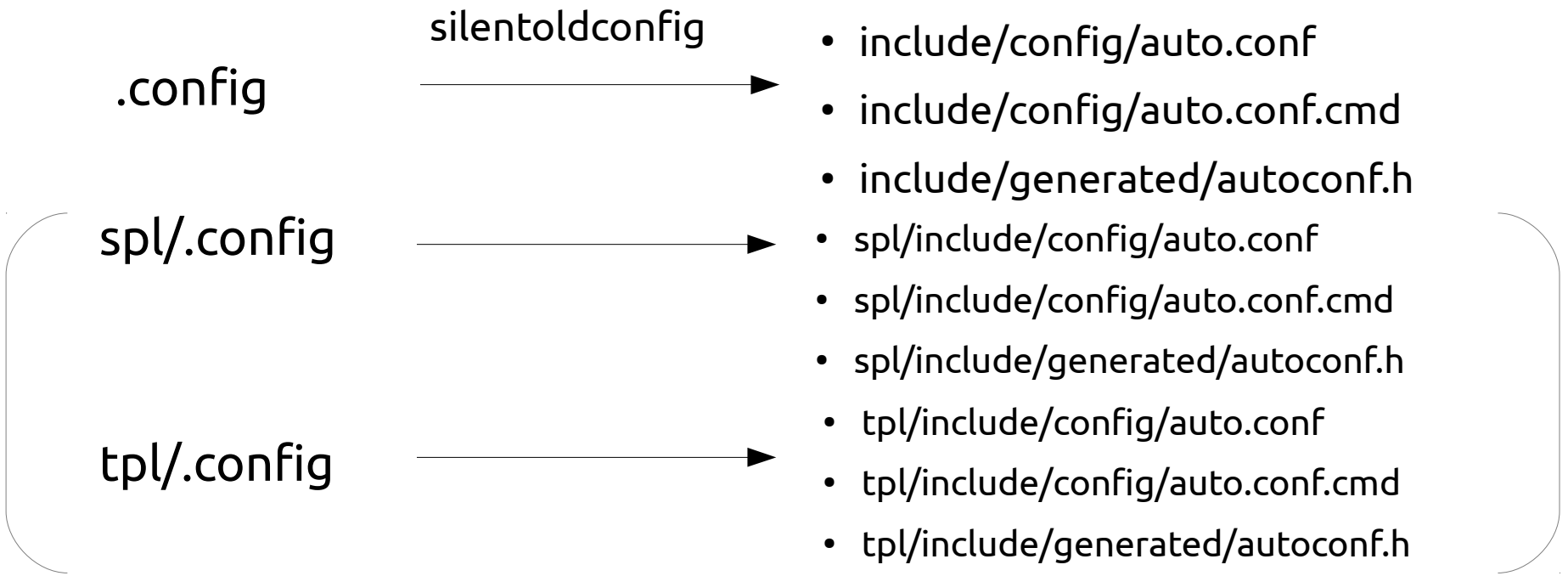
```
$ make <board>_defconfig
(or make <board>_config)
```

configs/<board>_defconfig → scripts/kconfig/conf →

- .config
- spl/.config
  (if CONFIG_SPL=y)
- tpl/.config
  (if CONFIG_TPL=y)

Configuration by Kconfig

(2) `$ make`



For support of old config
(will be removed in the future)

include/common.h
(include/configs/<board>.h)

scripts/Makefile.autoconf

- arch/${ARCH}/include/asm/arch
  => arch/${ARCH}/include/asm/arch-${SOC}
- include/config.h
- include/autoconf.mk
- include/autoconf.mk.dep
- [spl/include/autoconf.mk]
- [tpl/include/autoconf.mk]

Modifying .config

“make config” and friends create/modify “.config”

(menuconfig, gconfig, nconfig, xconfig)

```
$ make config
```

for .config

```
$ make spl/config
```

for spl/.config

(only when CONFIG_SPL is enabled)

```
$ make tpl/config
```

for tpl/.config

(only when CONFIG_TPL is enabled)

Where has boards.cfg gone?

Before Kconfig,
"boards.cfg" was the entry point of the build

Kconfig does not use boards.cfg

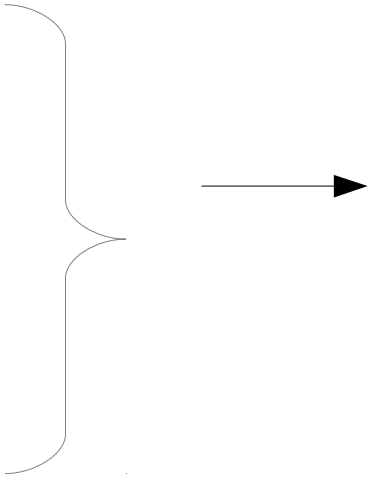
"configs/*_defconfig" is a new entry point

How was boards.cfg converted to Kconfig?

Where has info in boards gone?

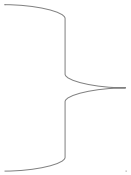
Fields in boards.cfg

- Arch
- CPU[:SPLCPU]
- SoC
- Vendor
- Board Name
- Config Name



→ Kconfig

- Target → File name of defconfig (configs/<target>_defconfig)
- Options → defconfig



- Status
- Maintainers

→ MAINTAINERS

defconfig format

Ex. 1

```
+S:CONFIG_ARM=y
+S:CONFIG_TEGRA=y
+S:CONFIG_TEGRA124=y
+S:CONFIG_TARGET_JETSON_TK1=y
CONFIG_DEFAULT_DEVICE_TREE="tegra124-jetson-tk1"
```

Ex. 2

```
CONFIG_FOO=100
S:CONFIG_FOO=200
T:CONFIG_FOO=300
ST:CONFIG_BAR=y
+S:CONFIG_BAZ=y
+T:CONFIG_QUX=y
+ST:CONFIG_QUUX=y
```

prefix	valid for
None	Normal
S:	SPL
T:	TPL
ST:	SPL, TPL
+S:	Normal, SPL
+T:	Normal, TPL
+ST:	Normal, SPL, TPL

boards.cfg is a generated file

The core build system itself does not need boards.cfg, but...

- Useful to browse the supported boards
- Necessary for MAKEALL and buildman

To generate it , run

```
$ tools/genboardscfg.py
```

Next Plan

Unspecialize SPL

SPL implementation is too special

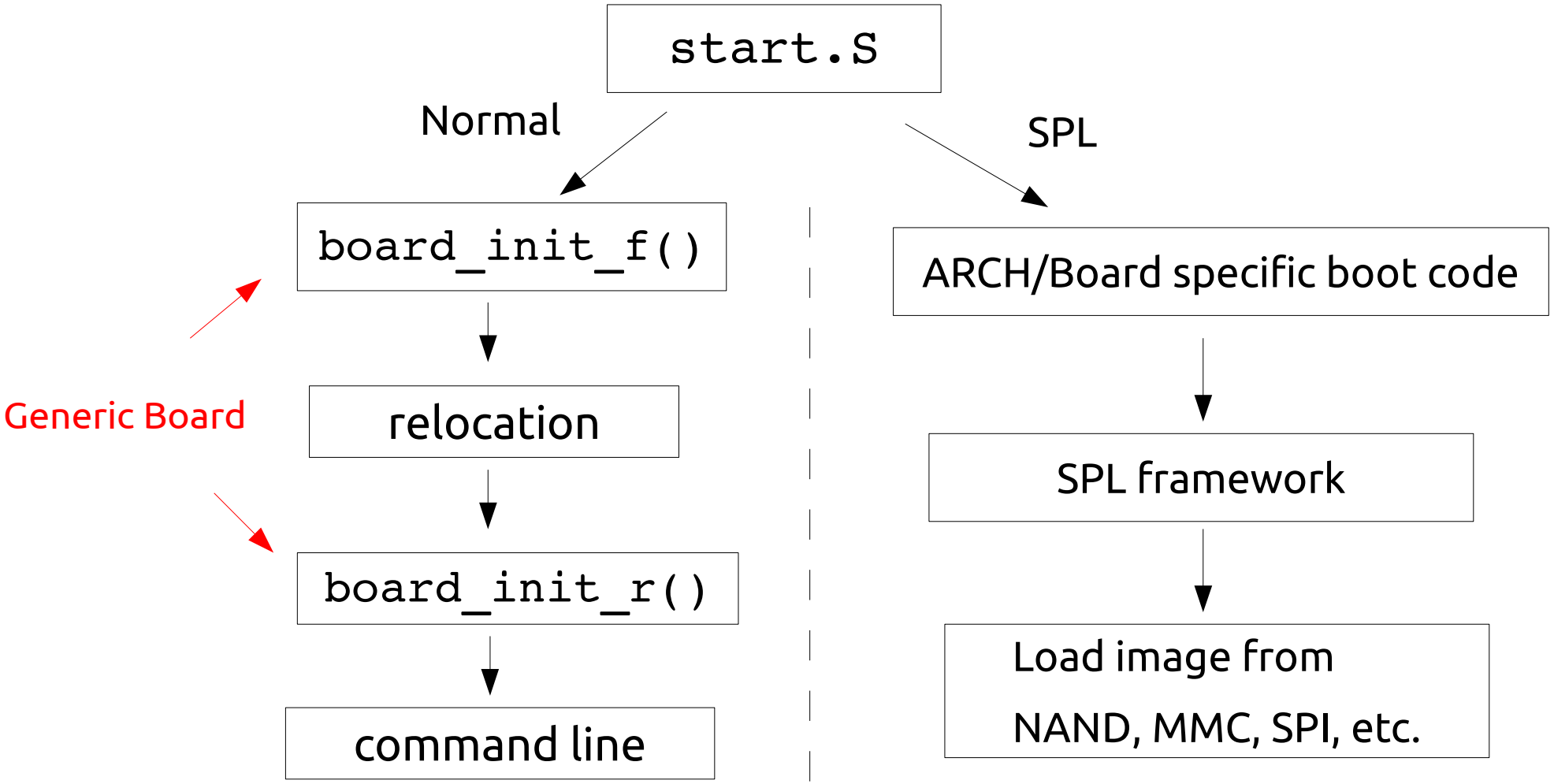
- Special Makefile (scripts/Makefile.spl)
- Special CONFIG (CONFIG_SPL_*)
- Special linker script (u-boot.lds)
- Special boot sequence (common/spl/spl.c)

Special CONFIGs to enable features in SPL:

scripts/Makefile.spl

```
libs-$(CONFIG_SPL_LIBCOMMON_SUPPORT) += common/
libs-$(CONFIG_SPL_LIBDISK_SUPPORT) += disk/
libs-$(CONFIG_SPL_I2C_SUPPORT) += drivers/i2c/
libs-$(CONFIG_SPL_GPIO_SUPPORT) += drivers/gpio/
libs-$(CONFIG_SPL_MMC_SUPPORT) += drivers/mmc/
libs-$(CONFIG_SPL_MPC8XXX_INIT_DDR_SUPPORT) += drivers/ddr/fsl/
libs-$(CONFIG_SPL_SERIAL_SUPPORT) += drivers/serial/
libs-$(CONFIG_SPL_SPI_FLASH_SUPPORT) += drivers/mtd/spi/
libs-$(CONFIG_SPL_SPI_SUPPORT) += drivers/spi/
   ...
```

Difference of boot sequence



Support

- CONFIG_OF_CONTROL
- CONFIG_FIT
- CONFIG_DM

etc.

Separate work for

- CONFIG_SPL_OF_CONTROL ?
- CONFIG_SPL_FIT ?
- CONFIG_SPL_DM ?
- CONFIG_SPL_GENERIC_BOARD ?

Some factors that differentiate Normal and SPL

- Normal
- SPL
- TPL

→ Generic case

	Normal	SPL
Relocation	ON	OFF
Drivers	Many & Full feature	Minimal & simple
Useful commands	Many	None
Goal	Invoke command line	Load another image

For example,
CONFIG_RELOCATION rather than CONFIG_SPL_BUILD

TODO items (please help!)

- Deprecate CONFIG_SYS_EXTRA_OPTIONS

```
CONFIG_SYS_EXTRA_OPTIONS="CUBIEBOARD2,AXP209_POWER"
```

- Don't use plain defines

```
#define CONFIG_SYS_BAUDRATE_TABLE     \
                {19200, 38400, 57600, 115200}
```

- Hierarchize board select menus

  Now a hundred boards in one menu

- Reduce the number of defconfigs per board

  ex. configs/MPC8313ERDB_33_defconfig
  configs/MPC8313ERDB_66_defconfig
  configs/MPC8313ERDB_NAND_33_defconfig
  configs/MPC8313ERDB_NAND_66_defconfig

Conclusion

We've got the nice infrastructure

but, lots of cleanups to do!

Please give me a hand!!

Thank you!

Questions?