



Embedded Software

Bootloader

Young-woo Lee

System IC Design Technology Laboratory

Bootloader

시스템의 하드웨어를 초기화하고 운영체제의 커널을 메모리에 실행시키는 시스템 프로그램

❖ 부트로더의 위치

- 일반적으로 시스템 메모리의 물리 주소 '0'번지에 위치
 - 0x0000_0000 : Reset exception

❖ 부트로더의 기능 : **Startup, Monitoring mode, OS Boot**

- 하드웨어 초기화
 - Clock 설정, memory(SDRAM) 설정, 인터럽트 disable
 - 시리얼포트 초기화
 - 네트워크 초기화
- 커널과 file system 적재
 - 커널과 root file system을 메모리에 적재하여 Linux 실행

Bootloader

❖ U-boot

- Universal Bootloader, Power PC와 ARM에 기반을 둔 임베디드 보드를 위한 부트로더
- ppcboot + armboot
- PPC와 ARM에 기반을 둔 임베디드용 Bootloader
- 다양한 cpu 지원 (PPC, ARM, MIPS, SH, x86...)
- Open Source
- 다른 CPU와 명령어 체계를 사용하므로, 다른 Platform에 적용이 용이하다.

Bootloader

❖ U-boot 주요 기능

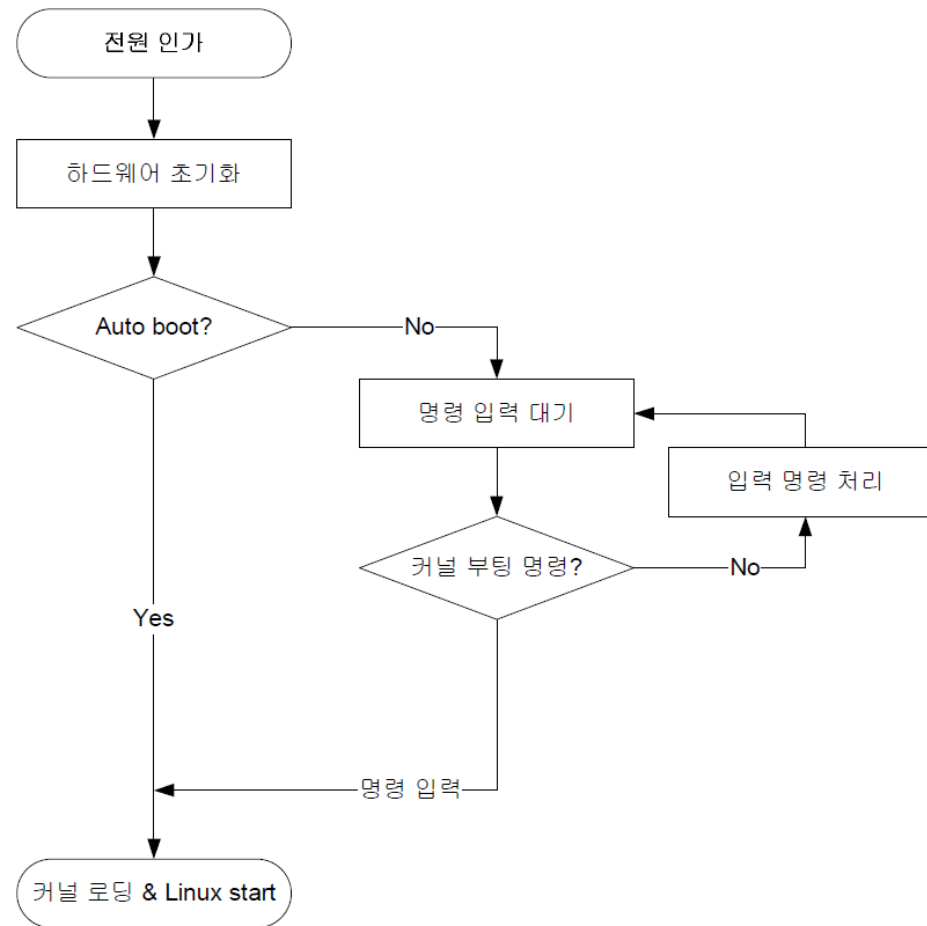
- Initialize base hardware
 - CPU speed
 - memory timings
 - Detect machine type (passed to Kernel by r1)
- Initialize device
 - Init UART to be Kernel console
(passed to Kernel by tagged list, kernel boot option)
- Copy Kernel and RFS image to RAM *RFS: Root File System
 - Kernel을 위한 연속된 physical memory의 블록을 준비한다. 또한 RFS을 위한 memory도 준비한다 (인접할 필요는 없다).
 - Copy (또는 download) 한 후에 Kernel과 RFS를 각각의 연속된 공간에 압축을 푼다.
- Setup the Kernel Tagged List (보드 정보)
 - ppc : structure
 - arm: physical ram 시작번지 + 0x100 (0xA000_0100)

Bootloader

❖ U-boot 주요 기능

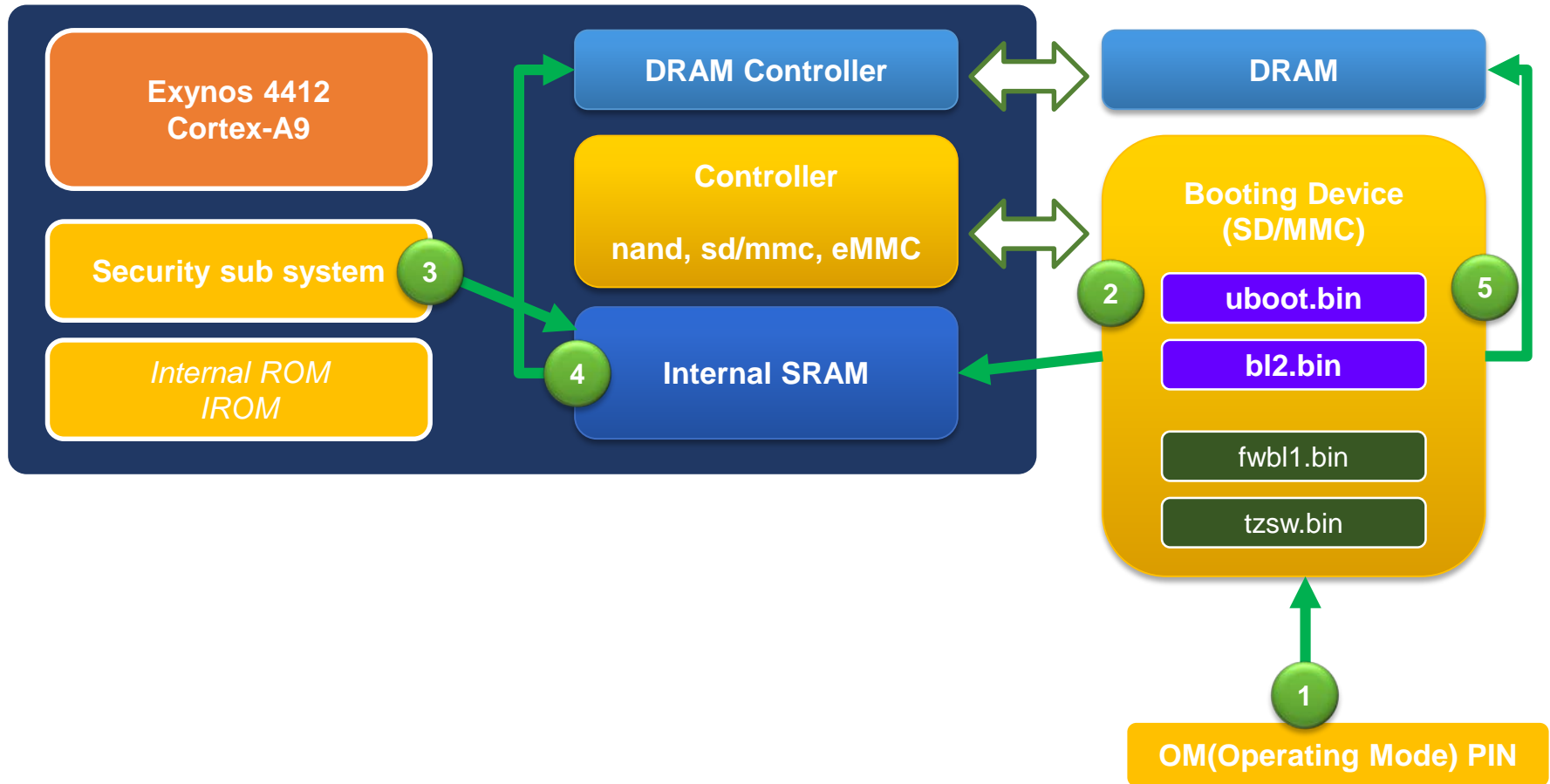
- Calling the Linux Kernel with following settings
 - CPU register settings
 - ✓ R0 = 0,
 - ✓ R1 = machine architecture number (the **MACH_TYPE_XXX** in kernel)
(linux/include/asm/mach-types.h)
Linux/arch/arm/tools/mach-types의 것과 일치해야 한다.
 - ✓ R2 = base address of ATAG list
(system RAM안에 있는 tagged list의 physical address)
 - CPU mode
 - ✓ All forms of interrupts must be disabled (IRQs, FIQs)
 - ✓ CPU must be in SVC mode
 - Cache, MMU *MMU: Memory Management Unit
 - ✓ MMU must be off
 - ✓ D-cache must be off and no any stale data (I-Cache can be on/off)

❖ 부트로더 실행 순서



Booting Sequence

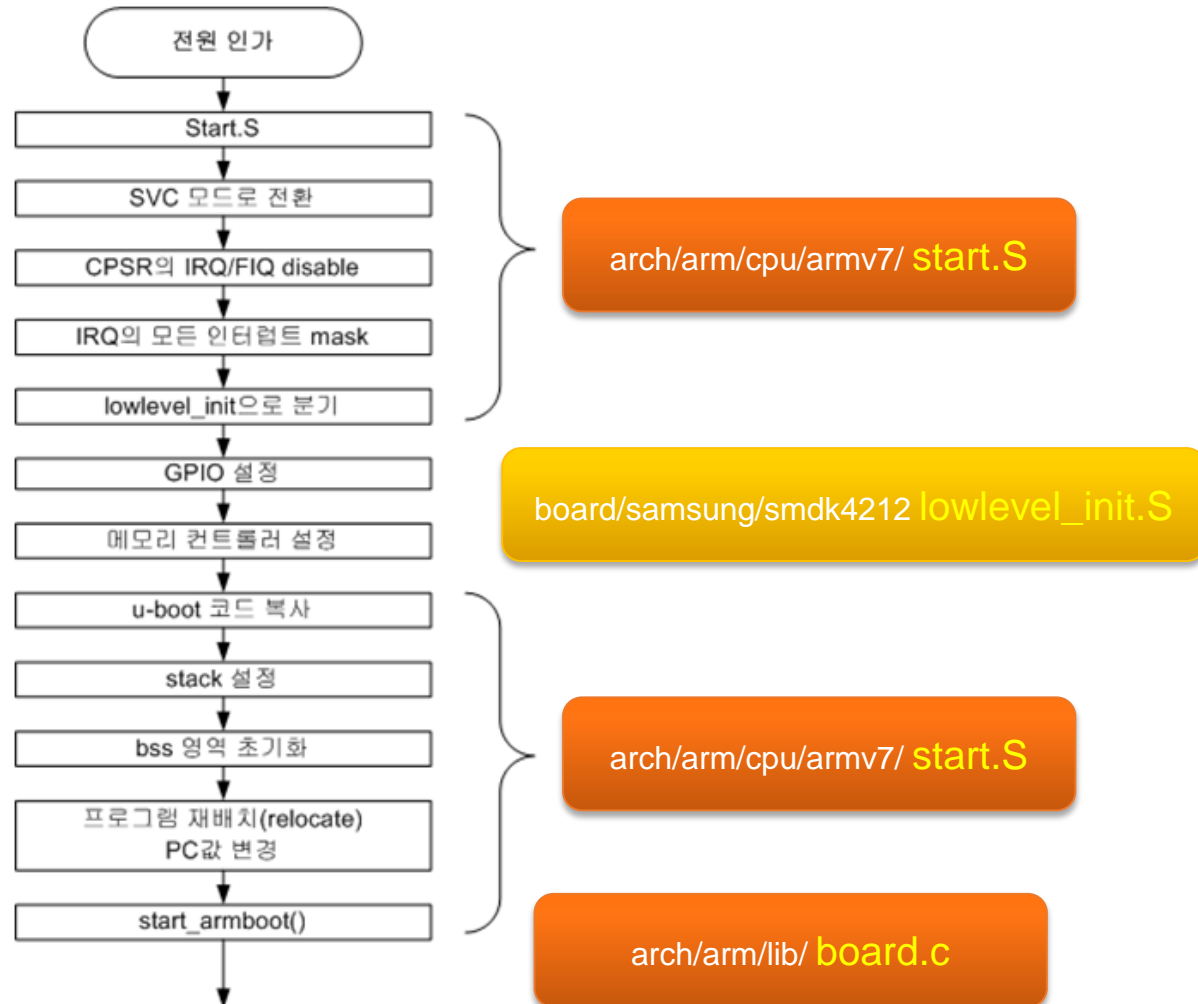
❖ u-boot booting 과정



Booting Sequence

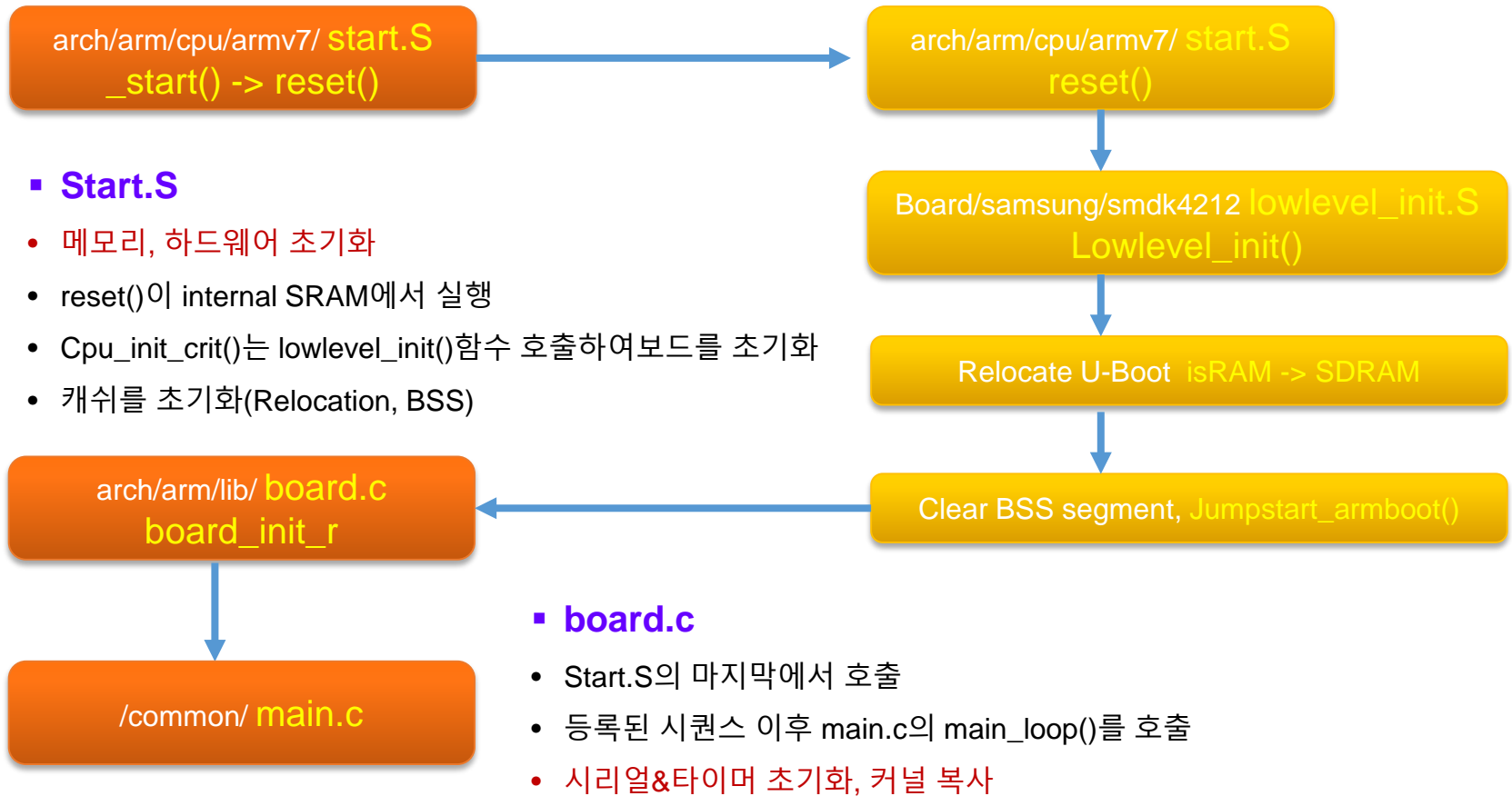
❖ u-boot 흐름 분석

1. Exception vector table 선언
2. Mode change(SVC)와 인터럽트 disable(IRQ, FIQ)
3. Watchdog timer disable
4. Interrupt disable by interrupt controller
5. PLL setting
6. Cache flush
7. Disable MMU, Disable D-Cache
8. U-boot code를 ram으로 복사
9. BSS영역 clear
10. Stack pointer 초기화



Booting Sequence

❖ u-boot 흐름 분석



Bootloader Files

❖ 부트로더 주요 파일

1. U-boot.lds

- u-boot의 링커스크립트 파일, 즉 ID가 u-boot를 링킹시에 이 lds파일을 참조하여 executable ELF를 생성
- LD(Loader&Linker)의 input으로 주어져서 object파일을 생성하는데 규칙을 제공

2. start.S

- 인터럽트 disable
- 벡터 테이블 설정
- 레지스터 설정(CPU 클럭, 메모리 설정)
- 부트로더를 RAM으로 복사
- Stack Point 설정
- 메모리상의 부트로더로 제어권 이양

3. main.c

- 모든 함수는 이곳에서 관리
- 해당 함수는 각 해당 C파일의 함수를 호출하여 수행

Bootloader Files

❖ U-boot.lds 흐름 분석 `board/samsung/smdk4212/u-boot.lds`

U-boot.lds에서 ENTRY에 지정된 "_start"함수로(start.S) u-boot프로그램이 시작된다. U-boot.lds파일은 LD (loader&Linker)의 Input으로 주어져서, object파일을 생성하는데 규칙을 제공한다.

```
root@ubuntu: /4412_Linux/bootloader/uboot_4412/board/samsung/smdk4212
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston,
* MA 02111-1307 USA
*/

OUTPUT_FORMAT("elf32-littlearm", "elf32-littlearm", "elf32-littlearm")
OUTPUT_ARCH(arm)
ENTRY(_start)
SECTIONS
{
    . = 0x00000000;
    . = ALIGN(4);
    .text :
    {
        arch/arm/cpu/armv7/start.o      (.text)
        board/samsung/smdk4212/libsmk4212.o (.text)
        arch/arm/cpu/armv7/exynos/libexynos.o (.text)
        *(.text)
    }
}
```

43,1-8 32%

Bootloader Files

❖ U-boot.lds 흐름 분석 `board/samsung/smdk4212/u-boot.lds`

- SECTIONS를 보면 text, rodata, data, got, mmudata, bss라는 section들이 정의되어 있다.
 - .text : 실행할 프로그램 코드 영역
 - .rodata : read-only data 영역 (const 등으로 지정된 데이터)
 - .data : initialized data 영역
 - .got : global offset table 영역
 - .mmudata : MMU data 영역
 - .bss : uninitialized data 영역
- 특수한 링커 변수 dot '.' 는 항상 현재 출력 address point를 담고 있다.
address point 는 출력 섹션의 크기만큼 증가한다.
- '*'는 어떤 파일명에도 대응한다. `*(.text)'는 모든 입력파일의 모든 입력 섹션 '.text'을 의미한다.
- 프로그램 코드는 0x00000000에서 시작해서 4byte단위로 정렬된 text section에 놓여질 것이다.
- U-boot의 시작은 Entry point에 선언된 _start부터 시작된다.
- _start는 (TOPDIR)/cpu/s5pc11x/start.S에 정의되어 있다.
- TEXT_BASE에의해 Linker수행 시 symbol들은 상대적 주소를 갖는다.
- Power가 on 한 후 0x00번지(즉,flash)에서 시작하여 memory초기화를 거쳐 flash의 내용을 dram에 relocate 하면 비로써 dram 에서 동작 하게 된다.
- Symbol들은 모두 TEXT_BASE의 상대적 주소 값을 가지고 있으므로, dram에 relocate전에는 offset branch 명령만 사용 해야 한다.(B,BL,ADR)

Bootloader Files

❖ Start.S 코드 `arch/arm/cpu/armv7/start.S`

.globl _start	Flash start, Linker에게 "_start" symbol 처리 할 수 있도록 export
<pre> _start: b reset ldr pc, _undefined_instruction ldr pc, _software_interrupt ldr pc, _prefetch_abort ldr pc, _data_abort ldr pc, _not_used ldr pc, _irq ldr pc, _fiq </pre>	<p>ARM에서 발생할 수 있는 exception으로 상황에 대한 처리 routine들로 exception이 발생하면 해당하는 핸들러로 branch할 것이다.</p> <p>exception의 처음인 0x00000000번지이다. 이 명령에 의해 reset로 branch한다. ldr pc, exception vector 설정</p>
<pre> _undefined_instruction: .word undefined_instruction _software_interrupt: .word software_interrupt _prefetch_abort: .word prefetch_abort _data_abort: .word data_abort _not_used: .word not_used _irq: .word irq _fiq: .word fiq _pad: .word 0x12345678 </pre>	exception vector의 위치 정의 word로 정의 한것

Bootloader Files

❖ Start.S 코드 `arch/arm/cpu/armv7/start.S`

<pre>.globl _TEXT_BASE _TEXT_BASE: .word CONFIG_SYS_TEXT_BASE</pre>	<p>TEXT_BASE는 <code>/board/samsung/smdk4212/config.mk</code>에 정의되어 있다. 정의되어 있는 값은 <code>0xc3e00000</code>이다. <code>0xc3e00000</code>은 H-SMART4412TKU의 SDRAM영역에 포함된다.</p>
<pre>.globl _bss_start_ofs _bss_start_ofs: .word __bss_start - _start .globl _bss_end_ofs _bss_end_ofs: .word _end - _start</pre>	<p><code>board/samsung/smdk4212/u-boot.lds</code>에서 참조</p>
<pre>reset: /* * set the cpu to SVC32 mode */ mrs r0, cpsr bic r0, r0, #0x1f orr r0, r0, #0xd3 msr cpsr,r0</pre>	<ul style="list-style-type: none"> • Clock setting, Memory setting • <code>mrs</code> : cpsr의 모든 비트를 r0레지스터로 복사 • <code>bic</code> : 하위 5bit clear, 즉 mode setting부분 clear • <code>orr</code> : interrupt을 disable, supervisor mode • <code>msr</code> : cpsr에 변경된 값 update

Bootloader Files

❖ Start.S 코드 `arch/arm/cpu/armv7/start.S`

cpu_init_crit

```
bl cache_init
```

```
/*
```

```
 * Invalidate L1 I/D
```

```
*/
```

```
mov    r0, #0                @ set up for MCR
mcr     p15, 0, r0, c8, c7, 0 @ invalidate TLBs
mcr     p15, 0, r0, c7, c5, 0 @ invalidate icache
```

cpu_init_crit = Memory initialize, 중요한 registers 및 memory timing등을 setup한다.

mov = flush, 기억장치 부분의 내용을 비운다.

mcr = 프로세서 15의 8번레지스터에 r0값, 초기화

Bootloader Files

❖ Start.S 코드 필요성

- 스택의 초기화
- 임베디드 시스템의 기본적인 초기화 작업
- 메모리, 스택, 전역변수 초기화

❖ Exception on Handler

0x00	Branch to Handler	Reset
0x04	Branch to Handler	Undefined Instruction
0x08	Branch to Handler	Software Interrupt
0x0c	Branch to Handler	Prefetch Memory Abort
0x10	Branch to Handler	Data Memory Abort
0x14	Branch to Handler	Reserved
0x18	Branch to Handler	Normal Interrupt
0x1c	Branch to Handler	Fast Interrupt
Interrupt Handler		

```

HandleIRQ      #      4

AREA   Init, CODE, READONLY
ENTRY
B      ResetHandler
B      HandlerUndef   ; Undefined
B      HandlerSWI     ; SWI interrupt
B      HandlerPabort  ; Pre-fetch Abort
B      HandlerDabort  ; Data Abort
B      .              ; Reserved
B      HandleIRQ      ; IRQ
B      HandleFIQ      ; FIQ
  
```

```

HandlerFIQ
SUB     sp, sp, #4
STMFD  sp!, {r0}
LDR     r0, =HandleLabel
LDR     r0, [r0]
STR     r0, [sp,#4]
LDMFD  sp!, {r0,pc}

...
  
```

```

^ _JSR_STARTADDRESS
HandleReset    #      4
HandleUndef    #      4
HandleSWI      #      4
HandlePabort   #      4
HandleDabort   #      4
HandleReserved #      4
HandleIRQ      #      4
HandleFIQ      #      4
  
```

Bootloader 실습

부트로더 흐름분석

U-Boot Build 환경

- 플랫폼 환경에서 리눅스 시스템을 구축하기 위해서는 가장 먼저 부트로더 시스템 프로그램이 필요

실습을 통해 부트로더 설치를 위한 Build시스템을 구축하고, 부트로더 흐름을 분석해보자.

[실습] U-Boot Build 환경 만들기

- U-Boot 압축해제

```
# cd /Smart4412Linux/Development/Source/bootloader/
# tar zxvf uboot_4412_linux.tar.gz
```

```
root@ubuntu: /Smart4412Linux/Development/Source/bootloader
root@ubuntu: /# cd /Smart4412Linux/Development/Source/bootloader/
root@ubuntu: /Smart4412Linux/Development/Source/bootloader# ls
uboot_4412_linux.tar.gz
root@ubuntu: /Smart4412Linux/Development/Source/bootloader# tar zxvf uboot_4412_linux.tar.gz

root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412
uboot_4412/examples/api/Makefile
uboot_4412/examples/api/.gitignore
uboot_4412/examples/api/demo.c
uboot_4412/examples/api/libgenwrap.c
uboot_4412/examples/api/glue.h
uboot_4412/examples/api/crt0.S
uboot_4412/examples/api/glue.c
uboot_4412/boards.cfg
root@ubuntu: /Smart4412Linux/Development/Source/bootloader# ls
uboot_4412 uboot_4412_linux.tar.gz
root@ubuntu: /Smart4412Linux/Development/Source/bootloader# cd uboot_4412
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# ls
api          common      CREDITS     examples    MAINTAINERS  mkconfig    post        snapshot.commit
arch         config.mk   disk        fs          MAKEALL      nand_spl    README     tools
board        COPYING     doc         include     Makefile     net         rules.mk
boards.cfg   COPYING.txt drivers     lib         mkb12        onenand_tpi sd_fuse
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412#
```

- ▣ 생성된 uboot_4412 디렉터리의 내용은 다음과 같다

디렉토리	내용
/api	API 관련 코드 및 예제
/board	board에 의존적인 파일
/common	architecture에 독립적인 파일
/disk	.Code for disk drive partition handling
/doc	uboot 관련 문서
/drivers	외부장치의 driver파일들
/examples	uboot을 위한 test실행 파일
/fs	uboot에서 지원하는 file system관련 파일
/include	header file, /configs(각보드설정파일)
/lib_	architecture관련 라이브러리 파일
/net	network 관련 파일
/post	Power On Self Test
/sd_fusing	SD를 이용한 부팅 관련 파일
/tools	Tools to build S-Record or U-Boot images, etc.

부트로더 흐름분석

- uboot_4412 디렉터리 안에 존재하는 부트로더의 주요 파일을 열어서 내용을 살펴보도록 한다.
- * u-boot.lds (uboot_4412/board/samsung/smdk4212/u-boot.lds)
 - u-boot의 링커스크립트 파일, 즉 ID가 u-boot를 링킹시에 이 lds파일을 참조하여 executable ELF를 생성
 - LD(Loader&Linker)의 input으로 주어져서 object파일을 생성하는데 규칙을 제공
- * start.S (uboot_4412/arch/arm/cpu/armv7/start.S)
 - 인터럽트 disable
 - 벡터 테이블 설정
 - 레지스터 설정(CPU 클럭, 메모리 설정)
 - 부트로더를 RAM으로 복사
 - Stack Point 설정
 - 메모리상의 부트로더로 제어권 이양
- * main.c (uboot_4412/common/main.c)
 - 모든 함수는 이곳에서 관리
 - 해당 함수는 각 해당 C파일의 함수를 호출하여 수행

부트로더 Build

U-Boot Build

- Build 환경이 구축되었으면 H-SMART4412TKU의 부트로더 Build를 적용하여 U-Boot 이미지를 생성
- U-Boot Build적용은 다음 명령을 통해 이루어짐

```
# make distclean  
# make smdk4412_config  
# make
```

U-Boot 디렉터리로 이동하여 make명령을 통해 Build작업을 실행

[실습] U-Boot Build하기

- make distclean

기존에 작업하였던 object, config를 삭제하고 초기화상태로 만든다.

```
# make distclean
```

```
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412  
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# ls  
api          common      CREDITS     examples    MAINTAINERS  mkconfig    post        snapshot.commit  
arch         config.mk   disk        fs          MAKEALL      nand_spl    README     tools  
board        COPYING    doc         include     Makefile     net         rules.mk  
boards.cfg   COPYING.txt drivers     lib         mkbl2        onenand_ipl sd_fuse  
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# make distclean  
awk '(NF && $1 !~ /^#/) { print $1 ": " $1 "_config; $(MAKE)" }' boards.cfg > .boards.depend  
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412#
```

부트로더 Build

▫ make smdk4412_config

'/include/configs/smdk4412_config.h'를 사용하여 H-SMART4412TKU에 대한 환경설정을 변경한다.

```
# make smdk4412_config
```

```
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# ls
api          common      CREDITS     examples    MAINTAINERS  mkconfig    post        snapshot.commit
arch         config.mk   disk        fs          MAKEALL      nand_spl    README     tools
board        COPYING    doc         include     Makefile     net         rules.mk
boards.cfg   COPYING.txt drivers     lib         mkb12        onenand_ipl sd_fuse
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# make distclean
awk '(NF && $1 !~ /^#/)' { print $1 ": " $1 "_config; $(MAKE)" }' boards.cfg > .boards.depend
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# make smdk4412_config
awk '(NF && $1 !~ /^#/)' { print $1 ": " $1 "_config; $(MAKE)" }' boards.cfg > .boards.depend
Configuring for smdk4412 board...
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412#
```

▫ make

```
# make
```

```
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# ls
api          common      CREDITS     examples    MAINTAINERS  mkconfig    post        snapshot.commit
arch         config.mk   disk        fs          MAKEALL      nand_spl    README     tools
board        COPYING    doc         include     Makefile     net         rules.mk
boards.cfg   COPYING.txt drivers     lib         mkb12        onenand_ipl sd_fuse
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# make distclean
awk '(NF && $1 !~ /^#/)' { print $1 ": " $1 "_config; $(MAKE)" }' boards.cfg > .boards.depend
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# make smdk4412_config
awk '(NF && $1 !~ /^#/)' { print $1 ": " $1 "_config; $(MAKE)" }' boards.cfg > .boards.depend
Configuring for smdk4412 board...
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# make
Generating include/autoconf.mk
```

부트로더 Build

- Build가 정상적으로 완료되면 u-boot.bin 파일과 bl2.bin 파일을 확인할 수 있다.

```

root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412
video.o drivers/watchdog/libwatchdog.o fs/cramfs/libcramfs.o fs/ext2/libext2fs.o fs/fat/libfat.o fs/fd
os/libfdos.o fs/jffs2/libjffs2.o fs/reiserfs/libreiserfs.o fs/ubifs/libubifs.o fs/yaffs2/libyaffs2.o l
ib/libfdt/libfdt.o lib/libgeneric.o lib/lzma/liblzma.o lib/lzo/liblzo.o net/libnet.o post/libpost.o bo
ard/samsung/smdk4212/libsmk4212.o --end-group /Smart4412Linux/Development/Source/bootloader/uboot_441
2/arch/arm/lib/eabi_compat.o -L /usr/local/CodeSourcery/Sourcery_G++_Lite/bin/./lib/gcc/arm-none-eabi
/4.4.1 -lgcc -Map u-boot.map -o u-boot
/usr/local/CodeSourcery/Sourcery_G++_Lite/bin/arm-none-eabi-objcopy -O srec u-boot u-boot.srec
/usr/local/CodeSourcery/Sourcery_G++_Lite/bin/arm-none-eabi-objcopy --gap-fill=0xff -O binary u-boot u
-boot.bin
./mkb12 u-boot.bin bl2.bin 14336
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412# ls
api          common      disk        include     mkb12       post         System.map  u-boot.map
arch         config.mk   doc         lib         mkconfig    README       tools       u-boot.srec
bl2.bin      COPYING     drivers     MAINTAINERS nand_spl    rules.mk     u-boot     u-boot.bin
board        COPYING.txt examples    MAKEALL     net         sd_fuse      u-boot.lds
boards.cfg   CREDITS     fs          Makefile    onenand_ipl snapshot.commit
root@ubuntu: /Smart4412Linux/Development/Source/bootloader/uboot_4412#

```

- 컴파일이 정상적으로 되지 않는 경우 실제 설치되어있는 컴파일러의 설치 경로와 부트로더의 config.mk파일을 확인한다.

```
# vi arch/arm/config.mk
```

- 하기 캡처 사진과 함께 고찰을 포함한 보고서 작성
 - ✓ [20페이지] 부트로더 주요파일 3가지 (u-boot.lds, start.S, main.c) 내용 일부
 - ✓ [23페이지] 부트로더 빌드 후 생성된 파일 2가지 (u-boot.bin, bl2.bin)

Any Questions?

Young-woo Lee
E-mail. ylee@jnu.ac.kr