Head.S 분석자료

16기 A조

Head.s @0

X0	DTB (PHY ADDRERSS)			
X1	00000000_00000000	X16		
X2	00000000_00000000	X17		
X3	00000000_00000000	X18		
X4		X19		
X5		X20		
X6		X21		
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30		LR

bl preserve_boot_args

X0	DTB (PHY ADDRERSS)			
X1	00000000_00000000	X16		
X2	00000000_00000000	X17		
X3	00000000_00000000	X18		
X4		X19		
X5		X20		
X6		X21		
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR

Head.s @2

```
mov     x21, x0
```

X0	**DTB (PHY ADDRERSS)**			
X1	**00000000_00000000**	X16		
X2	**00000000_00000000**	X17		
X3	**00000000_00000000**	X18		
X4		X19		
X5		X20		
X6		X21	**DTB (PHY ADDRERSS)**	
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR

```
adr_l   x0, boot_args

u64 __cacheline_aligned boot_args[4];
```

X0	&boot_args			
X1	00000000_00000000	X16		
X2	00000000_00000000	X17		
X3	00000000_00000000	X18		
X4		X19		
X5		X20		
X6		X21	DTB (PHY ADDRERSS)	
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR

```
stp     x21, x1, [x0]
stp     x2, x3, [x0, #16]

u64 __cacheline_aligned boot_args[4];

boot_args[0] = x21;     // DTB
boot_args[1] = x1;      // 0
boot_args[2] = x2;      // 0
boot_args[3] = x3;      // 0
```

X0	&boot_args			
X1	00000000_00000000	X16		
X2	00000000_00000000	X17		
X3	00000000_00000000	X18		
X4		X19		
X5		X20		
X6		X21	DTB (PHY ADDRERSS)	
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR

dmb sy

DMB
Data Memory Barrier.

SY
Full system barrier operation. This is the default and can be omitted.

X0	&boot_args			
X1	00000000_00000000	X16		
X2	00000000_00000000	X17		
X3	00000000_00000000	X18		
X4		X19		
X5		X20		
X6		X21	DTB (PHY ADDRERSS)	
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR

boot_arg[]를 저장한 명령어가
Chche clear 명령어에 영향을
받지 않도록 Barrier를 사용한다.
(메모리 오더링 이슈는 1 core에서도 발생한다)

Head.s @6

mov x1, #0x20

4 x 8 = 32 = 0x20

X0	&boot_args			
X1	**00000000_00000020**	X16		
X2	**00000000_00000000**	X17		
X3	**00000000_00000000**	X18		
X4		X19		
X5		X20		
X6		X21	**DTB (PHY ADDRERSS)**	
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR

b __inval_dcache_area

__inval_dcache_area(void *addr, size_t len);

X0 – addr
X1 – len

Ensure that any D-cache lines are invalidated.

X0	&boot_args			
X1	**00000000_00000020**	X16		
X2	**00000000_00000000**	X17		
X3	**00000000_00000000**	X18		
X4		X19		
X5		X20		
X6		X21	**DTB (PHY ADDRERSS)**	
X7		X22		
X8		X23		
X9		X24		
X10		X25		
X11		X26		
X12		X27		
X13		X28		
X14		X29		FP
X15		X30	stext	LR