

2018.05.08 - Mon

노트북: SW

만든 날짜: 2018-05-08 오전 10:13

수정한 날짜: 2018-05-09 오전 12:23

작성자: 정상요

2018. 4. 13 금 – 37회차

과정 : TI, DSP, Xilinx Zynq FPGA, MCU 기반의 프로그래밍 전문가 과정

Prof. 이상훈
gcccompil3r@gmail.com
Stu. 정상용
fstopdg@gmail.com

- Cortex-R bootloader 분석

CCS 실행

External_LED -> source -> HL_sys_startup.c

_c_int00부터 시작

systemInit()이전까지(금요일 수업 내용)

2018.05.08

F3 : 진입, Alt + <- : 나오기, HL_sys_core.asm : asm분석(stack pointer 보이면), F2 : 정의 확인

+참고

TMS570-TRM, ARM Cortex-R5F Technical Reference Manual

Manual, Datasheet 참고시 Register이름이 바뀌었을 수도 있음.(알아서 센스있게 찾으면 됨)

systemInit()

-> setupPLL()

-> systemREG1 : 0xFFFFFFFF00U

-> systemREG1 -> CSDISSET = 0x00000002U | 0x00000004U -> 0b 0100 0010

2.5.1.11 Clock Source Disable Set Register (CSDISSET)

The CSDISSET register, shown in Figure 2-18 and described in Table 2-30, sets clock sources to the disabled state.

Figure 2-18. Clock Source Disable Set Register (CSDISSET) (offset = 34h)

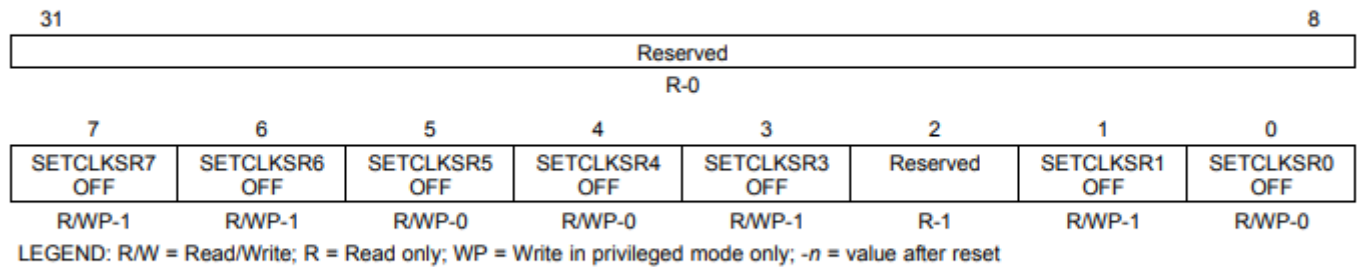


Table 2-30. Clock Source Disable Set Register (CSDISSET) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reads return 0. Writes have no effect.
7-3	SETCLKSR[7-3]OFF	0	Set clock source[7-3] to the disabled state. <i>Read:</i> Clock source[7-3] is enabled. <i>Write:</i> Clock source[7-3] is unchanged.
		1	<i>Read:</i> Clock source[7-3] is disabled. <i>Write:</i> Clock source[7-3] is set to the disabled state. Note: After a new clock source disable bit is set via the CSDISSET register, the new status of the bit will be reflected in the CSDIS register (offset 30h), the CSDISSET register (offset 34h), and the CSDISCLR register (offset 38h).
2	Reserved	1	Reads return 1. Writes have no effect.
1-0	SETCLKSR[1-0]OFF	0	Set clock source[1-0] to the disabled state. <i>Read:</i> Clock source[1-0] is enabled. <i>Write:</i> Clock source[1-0] is unchanged.
		1	<i>Read:</i> Clock source[1-0] is disabled. <i>Write:</i> Clock source[1-0] is set to the disabled state. Note: After a new clock source disable bit is set via the CSDISSET register, the new status of the bit will be reflected in the CSDIS register (offset 30h), the CSDISSET register (offset 34h), and the CSDISCLR register (offset 38h).

```
-> while(1) : wait
-> systemREG1 -> GBLSTAT = 0x301U
-> muxInit()
-> setupFlash()
-> trimP0()
-> mapClocks();
-> systemREG1 -> SYSPC1 = 0U
```

강의내용

-> trimP0

LPO : Low - Power Oscillator

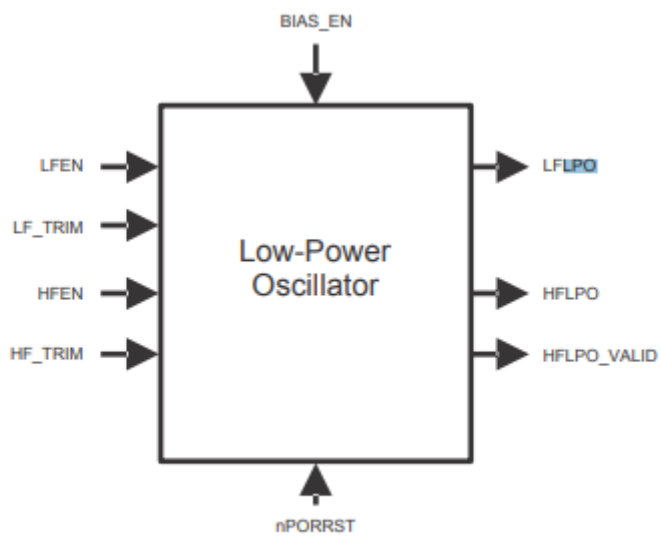


Figure 6-4. LPO Block Diagram

Table 2-49. LPO/Clock Monitor Control Register (LPOMONCTL) Field Descriptions (continued)

Bit	Field	Value	Description																																																																
12-8	HFTRIM		High-frequency oscillator trim value. This four-bit value is used to center the HF oscillator's frequency. Caution: This value should only be changed when the HF oscillator is not the source for a clock domain, otherwise a system failure could result. The following values are the ratio: f / fo in the F021 process. <table><tr><td>0</td><td>29.52</td></tr><tr><td>1h</td><td>34.24%</td></tr><tr><td>2h</td><td>38.85%</td></tr><tr><td>3h</td><td>43.45%</td></tr><tr><td>4h</td><td>47.99%</td></tr><tr><td>5h</td><td>52.55%</td></tr><tr><td>6h</td><td>57.02%</td></tr><tr><td>7h</td><td>61.46%</td></tr><tr><td>8h</td><td>65.92%</td></tr><tr><td>9h</td><td>70.17</td></tr><tr><td>Ah</td><td>74.55%</td></tr><tr><td>Bh</td><td>78.92%</td></tr><tr><td>Ch</td><td>83.17%</td></tr><tr><td>Dh</td><td>87.43%</td></tr><tr><td>Eh</td><td>91.75%</td></tr><tr><td>Fh</td><td>95.89%</td></tr><tr><td>10h</td><td>100.00% Default at Reset.</td></tr><tr><td>11h</td><td>104.09</td></tr><tr><td>12h</td><td>108.17</td></tr><tr><td>13h</td><td>112.32</td></tr><tr><td>14h</td><td>116.41</td></tr><tr><td>15h</td><td>120.67</td></tr><tr><td>16h</td><td>124.42</td></tr><tr><td>17h</td><td>128.38</td></tr><tr><td>18h</td><td>132.24</td></tr><tr><td>19h</td><td>136.15</td></tr><tr><td>1Ah</td><td>140.15</td></tr><tr><td>1Bh</td><td>143.94</td></tr><tr><td>1Ch</td><td>148.02</td></tr><tr><td>1Dh</td><td>151.80x</td></tr><tr><td>1Eh</td><td>155.50x</td></tr><tr><td>1Fh</td><td>159.35%</td></tr></table>	0	29.52	1h	34.24%	2h	38.85%	3h	43.45%	4h	47.99%	5h	52.55%	6h	57.02%	7h	61.46%	8h	65.92%	9h	70.17	Ah	74.55%	Bh	78.92%	Ch	83.17%	Dh	87.43%	Eh	91.75%	Fh	95.89%	10h	100.00% Default at Reset.	11h	104.09	12h	108.17	13h	112.32	14h	116.41	15h	120.67	16h	124.42	17h	128.38	18h	132.24	19h	136.15	1Ah	140.15	1Bh	143.94	1Ch	148.02	1Dh	151.80x	1Eh	155.50x	1Fh	159.35%
0	29.52																																																																		
1h	34.24%																																																																		
2h	38.85%																																																																		
3h	43.45%																																																																		
4h	47.99%																																																																		
5h	52.55%																																																																		
6h	57.02%																																																																		
7h	61.46%																																																																		
8h	65.92%																																																																		
9h	70.17																																																																		
Ah	74.55%																																																																		
Bh	78.92%																																																																		
Ch	83.17%																																																																		
Dh	87.43%																																																																		
Eh	91.75%																																																																		
Fh	95.89%																																																																		
10h	100.00% Default at Reset.																																																																		
11h	104.09																																																																		
12h	108.17																																																																		
13h	112.32																																																																		
14h	116.41																																																																		
15h	120.67																																																																		
16h	124.42																																																																		
17h	128.38																																																																		
18h	132.24																																																																		
19h	136.15																																																																		
1Ah	140.15																																																																		
1Bh	143.94																																																																		
1Ch	148.02																																																																		
1Dh	151.80x																																																																		
1Eh	155.50x																																																																		
1Fh	159.35%																																																																		
7-5	Reserved	0	Reads return 0. Writes have no effect.																																																																

중심주파수 :

Trim frequency : 원하는 주파수로 깎아내림 : 파형성형회로

-> mapClocks

차량용 통신 : CAN통신 : VCLKA1: 110MHZ가 다루는 것이 CAN통신

FlexRay : CAN통신의 일종 : CAN통신보다 조금 더 좋음 : VCLKA2가 다루는 것인 FlexRay

VCLK3 :

VCLK4 : ethernet으로 이용

EMIF 데이터시트에서 찾아볼것

VCLK3는 PWM을 컨트롤하는데
우리가 사용하는 RC용 모터는 20ms에 맞춰져있다. -> 50Hz
그러면 당연히 PLL을 건드려야 한다.

HTU : FPGA를 프로그램할 수 있음.(HTU를 사용하면 ps, ns를 감지가능)

PLL(Phase Loop Locked) : 위상이 정상화가 될 경우
clocked are locked : overshooted, underdamping이 지나고 정상화가 될 경우를 기다림

VCLKA가 안정화가 되면, CSVSTAT 가 1 -> 결국 하드웨어적인 부분

CSDIS : 현재 활성화된 CLK과 비활성화된 CLK을 구분(비활성화가 잘 구분가능 : 신호를 안정화시키는데 시간이 걸린다.)

발진기의 기본개념 : PLL

While : 안정화가 된지 안된지 확인하는 부분 -> 안정화가 되면 빠져나감

가끔보면 compiler의 쓸데없는짓, 멍청한 짓을 볼수가 있다.

RTI : Real Time Interrupt : 현재 사용x -> 컴파일러의 멍청한짓

CLK2CNTRL : VCLK3의 주파수를 결정

sysyemInit이 끝나면 CLK설정이 끝남(P.59에 정리되어있음)

coreEnableIrq

mpu : memory protection unit : linux와 다르게 mcu는 1:1 매핑을 한다 그러므로 주소만 알면 바로 해킹가능, 그것을 방지하기 위
해서 mpu를 생성

memory barrier : 최적화를 하지말라는 의미 : instruction scheduling 을 하지말라고

dsb 끝나기전까지는 cpu가 여러개여도 하지말라는 의미!

소프트웨어적으로는 semaphore등으로 블락이 가능하였지만, cpu가 여러개 일경우 다른 cpu가 접근하여서 문제가 생김 그래서
만든것이 dsb

데이터 동기화 장벽

isb

인스터럭션 동기화 장벽

이것은 data가 아니라 instruction level에서 동작

상용 - FPGA : 23, 26 장

ARM Cortex-A9 : 3, 4 장

TMS570 : 29, 30 장

수업내용정리

In CCS

External_LED

-> source

->HL_sys_startup.c

systemInit()

->trimLPO() 부터 시작

```
/* SourceId : SYSTEM_SourceId_002 */
/* DesignId : SYSTEM_DesignId_002 */
/* Requirements : HL_CONQ_SYSTEM_SR6 */
void trimLPO(void)
{
/* USER CODE BEGIN (4) */
/* USER CODE END */
/** @b Initialize Lpo: */
/** Load TRIM values from OTP if present else load user defined values */
/*SAFETYMCUSW 139 S MR:13.7 <APPROVED> "Hardware status bit read check" */
if(LPO_TRIM_VALUE != 0xFFFFU) /* It will be never happened */
{
    systemREG1->LPOMONCTL = ((uint32)((uint32)1U << 24U)
                                | LPO_TRIM_VALUE;
}
else
{
    systemREG1->LPOMONCTL = ((uint32)((uint32)1U << 24U)
                                | (uint32)((uint32)16U << 8U)
                                | 16U;
}
/* USER CODE BEGIN (5) */
/* USER CODE END */
}
```

trimLPO

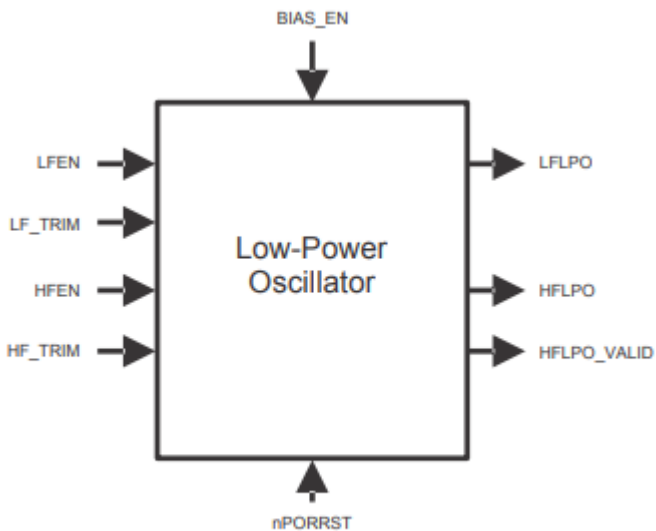


Figure 6-4. LPO Block Diagram

LPO from Cortex-R5 and Cortex-R5F Technical Reference Manual

-> if(LPO_TRIM_VALUE != 0xFFFFU)

```
#define LPO_TRIM_VALUE (((*(volatile uint32 *)0xF00801B4U) & 0xFFFF0000U)>>16U) /* 최상위 16bit  
만 뽑아내겠다 */
```

F008 01B4 먼저 찾기!!!

16U : 상위 16bit만 뽑아내겠다.

-> else

