Semiconductors

AN10331

Philips LPC2xxx family phase lock loop

Rev. 01 — 1 November 2004 **Application note**

Document information

Info	Content
Keywords	LPC2000, Phase Lock Loop Initialization
Abstract	Application information for the LPC2000 family Phase Lock Loop

Revision history

Rev	Date	Description
01	20041101	Initial version

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

This application note describes the different blocks of the Phase Lock Loop in the LPC2000 family of Philips ARM7 Microprocessors.

The application note is arranged in the following manner:

- Overview – a brief description of the Phase Lock Loop
- Activating the Phase Lock Loop – description of various registers and how to calculate the values that ensure the PLL works within the specified range of operation.
- Code example

2. Overview

The input frequency of the MCU is labeled f_{osc} (Oscillator Frequency). The label for the CPU clock is cclk. The labels for operating parameters for the Phase Lock Loop (PLL) block are MSEL - PLL Multiplier and PSEL - PLL Divider. The hardware can be run in 'PLL mode' where the f_{osc} must be between 10 MHz and 25 Mhz. At power on/reset, no software operation is required to enhance the performance of the CPU clock (cclk).The default value in the PLL registers is PLL mode turned off.

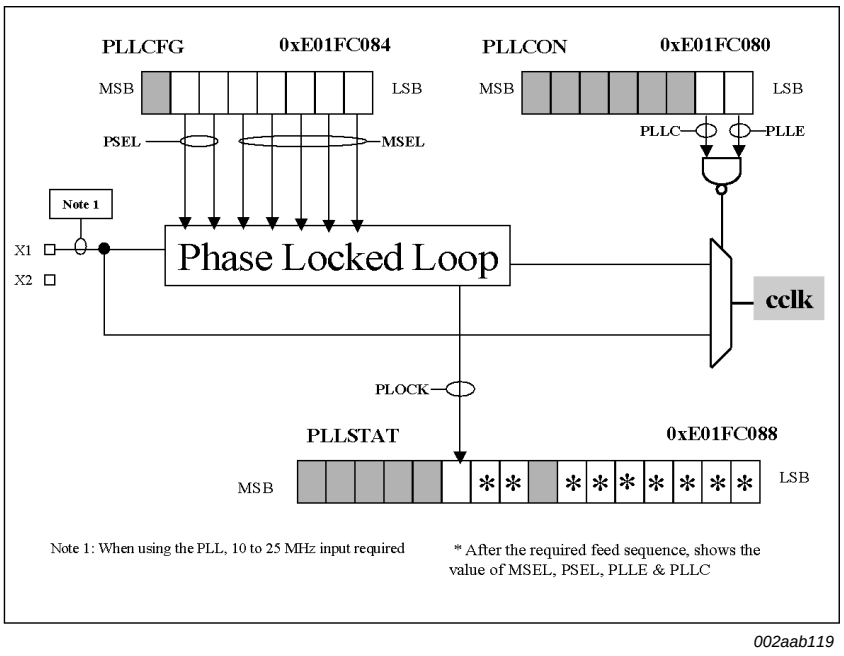

Fig 1. Phase lock loop block diagram and pseudo-logic.

3. Activating the Phase Lock Loop

As previously mentioned, using the PLL mode requires an f_{osc} between 10 Mhz and 25 Mhz. When using the PLL, there is an interaction between the f_{osc}, the desired CPU clock (cclk), parameters used by the PLL block and the operating range of the PLL current controlled oscillator (Fcco). The minimum cclk generated by the PLL is 10 Mhz.

For CPU clock less than 10 MHz use oscillation or slave mode.

Table 1: PLL labels

Label	Name	Min	Max
f_{osc}	oscillator frequency (slave)	1 Mhz	50 Mhz
	oscillator frequency (oscillation mode)	1 Mhz	30 Mhz
	oscillator frequency (PLL Mode)	10 Mhz	25 Mhz
Fcco	current controlled oscillator frequency	156 Mhz	320 Mhz
cclk	CPU clock	1Mhz	60 Mhz
MSEL	PLL multiplier value	0x0	0x5
PSEL	PLL divider	0x0	0x3

Table 2: PLL multiplier values

MSEL hex	Multiplier value decimal
0	1
1	2
2	3
3	4
4	5
5	6

Table 3: PLL divider values

PSEL Hex	Divider Values decimal
0	1
1	2
2	4
3	8

3.1 CPU performance

Calculations for desired CPU performance are:

CPU Clock = Oscillator Frequency × PLL Multiplier Value or

(1) cclk = f_{osc} * MSEL

Note the minimum and maximum values in Table 1. Table 2 converts the hex value for MSEL into the decimal multiplier value used in this calculation.

3.2 Ensure PLL compliance

The Current Controlled Oscillator Frequency (Fcco) uses values MSEL and PSEL to derive an output from the PLL Block. Calculations used for the PLL are:

Current Controlled Oscillator Frequency = CPU Clock × 2 × PLL Divider or

(2) Fcco = cclk*2*PSEL

Note the minimum and maximum values for Fcco influence the selection of PSEL. The tables below shows some calculations based on the above variables. In the first column is the input frequency to the MCU. Equation (1) results for the values in the row are shown in the cclk column. The four columns to the right of MSEL that have an 'x' mark represent the valid PSEL values that could be used for the cclk in column 2.

The columns to the right represent results from performing equation (2). The shaded blocks indicate calculated values that are within the parameter specifications of Fcco.

Table 4: Calculated operating values for 10 MHz to 13 MHz

f_{osc}	cclk	MSEL (hex)	PSEL (hex)					cclk*2	PSEL (decimal)			
			0	1	2	3			1	2	4	8
10	10	0				x		20	20	40	80	160
10	20	1			x	x		40	40	80	160	320
10	30	2			x			60	60	120	240	480
10	40	3		x	x			80	80	160	320	640
10	50	4		x				100	100	200	400	800
10	60	5		x				120	120	240	480	960
11	11	0				x		22	22	44	88	176
11	22	1			x			44	44	88	176	352
11	33	2			x			66	66	132	264	528
11	44	3		x				88	88	176	352	704
11	55	4		x				110	110	220	440	880
12	12	0				x		24	24	48	96	192
12	24	1			x			48	48	96	192	384
12	36	2			x			72	72	144	288	576
12	48	3		x				96	96	192	384	768
12	60	4		x				120	120	240	480	960
13	13	0				x		26	26	52	104	208
13	26	1			x			52	52	104	208	416
13	39	2		x	x			78	78	156	312	624
13	52	3		x				104	104	208	416	832

Table 5: Calculated operating values for 14 MHz to 20 MHz

f_{osc}	cclk	MSEL (hex)	PSEL (hex)					cclk*2	PSEL (decimal)			
			0	1	2	3			1	2	4	8
14	14	0				x		28	28	56	112	224
14	28	1			x			56	56	112	224	448
14	42	2		x				84	84	168	336	672
14	56	3		x				112	112	224	448	896
15	15	0				x		30	30	60	120	240
15	30	1			x			60	60	120	240	480
15	45	2		x				90	90	180	360	720
15	60	3		x				120	120	240	480	960
16	16	0				x		32	32	64	128	256
16	32	1			x			64	64	128	256	512
16	48	2		x				96	96	192	384	768
17	17	0				x		34	34	68	136	272
17	34	1			x			68	68	136	272	544
17	51	2		x				102	102	204	408	816
18	18	0				x		36	36	72	144	288
18	36	1			x			72	72	144	288	576
18	54	2		x				108	108	216	432	864
19	19	0				x		38	38	76	152	304
19	38	1			x			76	76	152	304	608
19	57	2		x				114	114	228	456	912
20	20	0			x	x		40	40	80	160	320
20	40	1		x	x			80	80	160	320	640
20	60	2		x				120	120	240	480	960

Table 6: Calculated operating values 21 MHz to 25 MHz

f_{osc}	cclk	MSEL (hex)	PSEL (hex)					cclk*2	PSEL (decimal)			
			0	1	2	3			1	2	4	8
21	21	0			x			42	42	84	168	336
21	42	1		x				84	84	168	336	672
22	22	0			x			44	44	88	176	352
22	44	1		x				88	88	176	352	704
23	23	0			x			46	46	92	184	368
23	46	1		x				92	92	184	368	736
24	24	0			x			48	48	96	192	384
24	48	1		x				96	96	192	384	768
25	25	0			x			50	50	100	200	400
25	50	1		x				100	100	200	400	800

3.3 Load the PLL configuration register

Insert the calculated values of PSEL and MSEL into your project. The example at the end of this application note uses information from row two of Table 4 where MSEL = 1 ($2 \times f_{osc}$) and PSEL = 2 (f_{osc} calculation = 160 MHz).

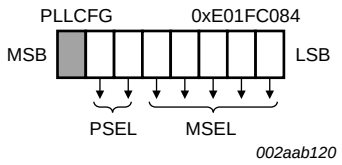

Fig 2. PLL configuration load.

3.4 Load the PLL control register

Assert the PLL enable (PLLE) bit. The PLLE bit tells the module to activate.

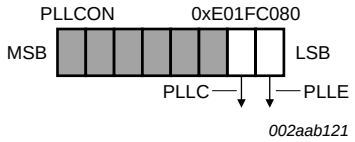

Fig 3. PLL control load.

3.5 Perform the validation sequence

The validation sequence consists of writing 0xAA to the PLLFEED register immediately followed by writing 0x55 to the PLLFEED register.

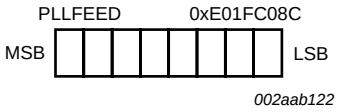

Fig 4. PLL validation.

3.6 Verify proper operation of the PLL

Check the PLL lock bit (PLOCK) in the PLL Status Register (PLLSTAT). It will set when the PLL has locked. You'll also be able to confirm the MSEL and PSEL values being used in generating the PLL output.

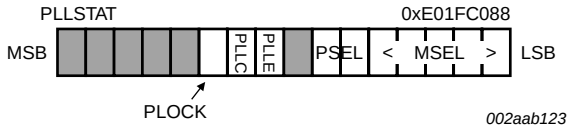

Fig 5. PLL verification.

3.7 Connect the PLL

Assert the PLL connect (PLLC) bit. The PLLC bit causes the cclk to be the PLL output.

3.8 Perform the validation sequence

The validation sequence consists of writing 0xAA to the PLLFEED register immediately followed by writing 0x55 to the PLLFEED register.

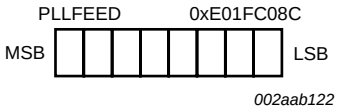

Fig 6. PLL connect.

4. Code example

```
#include <LPC21xx.H>
int main(void) {
/* PLL initialization
      * cclk = Fosc*MSEL
      * Fcco = cclk*2*PSEL
      *
      * PLLCFG                0xE01FC084
      *--------------------------------
      *I x I P I P I M I M I M I M I M I
      *--------------------------------
       x= do not use P= PSEL  M=MSEL        */
PLLCFG=0x22;
/*
      * PLLCON                0xE01FC080
      *--------------------------------
      *I x I x I x I x I x I C I E I
      *--------------------------------
       x= do not use C= PLLC  E=PLLE*/

PLLCON=0x1;

PLLFEED=0xAA;
PLLFEED=0x55;
/*
 * PLLSTAT                      0xE01FC088
 *-------------------------------*----------------------------------------
 I x I x I x I x I L I *cI *eI x I *pI *pI *mI *mI *mI *mI *mI
 *-------------------------------*----------------------------------------
x= do not use L = PLOCK *c=PLLC bit after feed *e=PLLE bit after feed
*p=PSEL bits after feed *m=MSEL bits after feed    */

while(!(PLLSTAT & 0x400)){}
;
PLLCON=0x3;

PLLFEED=0xAA;
PLLFEED=0x55;

while(1){};
}
```

5. References

[1] **AN10254 —** Philips LPC210x microcontroller family (9397 750 12478)

[2] **LPC2114/2124/2212/2214 User Manual —** 9397 750 13261

[3] **LPC2119/2129/2194/2292/2294 User Manual —** 9397 750 13262

6. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

7. Contents

Date of release: 1 November 2004
Document number: 9397 750 14142

Published in the U.S.A.