LPC2917/19

ARM9 microcontroller with CAN and LIN

Rev. 01 — 31 July 2008 **Product data sheet**

1. Introduction

1.1 About this document

This document lists detailed information about the LPC2917/19 device. It focuses on factual information like pinning, characteristics etc. Short descriptions are used to outline the concept of the features and functions. More details and background on developing applications for this device are given in the LPC2917/19 User manual (see Ref. 1). No explicit references are made to the User manual.

1.2 Intended audience

This document is written for engineers evaluating and/or developing systems, hard- and/or software for the LPC2917/19. Some basic knowledge of ARM processors and architecture and ARM968E-S in particular is assumed (see Ref. 2).

2. General description

2.1 Architectural overview

The LPC2917/19 consists of:

- An ARM968E-S processor with real-time emulation support
- An AMBA Advanced High-performance Bus (AHB) for interfacing to the on-chip memory controllers
- Two DTL buses (a universal NXP interface) for interfacing to the interrupt controller and the Power, Clock and Reset Control cluster (also called subsystem)
- Three ARM Peripheral Buses (APB - a compatible superset of ARM's AMBA advanced peripheral bus) for connection to on-chip peripherals clustered in subsystems.
- One ARM Peripheral Bus for event router and system control.

The LPC2917/19 configures the ARM968E-S processor in little-endian byte order. All peripherals run at their own clock frequency to optimize the total system power consumption. The AHB2APB bridge used in the subsystems contains a write-ahead buffer one transaction deep. This implies that when the ARM968E-S issues a buffered write action to a register located on the APB side of the bridge, it continues even though the actual write may not yet have taken place. Completion of a second write to the same subsystem will not be executed until the first write is finished.

2.2 ARM968E-S processor

The ARM968E-S is a general purpose 32-bit RISC processor, which offers high performance and very low power consumption. The ARM architecture is based on RISC principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed CISC. This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective controller core.

Amongst the most compelling features of the ARM968E-S are:

- Separate directly connected instruction and data Tightly Coupled Memory (TCM) interfaces
- Write buffers for the AHB and TCM buses
- Enhanced 16 × 32 multiplier capable of single-cycle MAC operations and 16-bit fixed-point DSP instructions to accelerate signal-processing algorithms and applications.

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. The ARM968E-S is based on the ARMv5TE five-stage pipeline architecture. Typically, in a three-stage pipeline architecture, while one instruction is being executed its successor is being decoded and a third instruction is being fetched from memory. In the five-stage pipeline additional stages are added for memory access and write-back cycles.

The ARM968E-S processor also employs a unique architectural strategy known as Thumb, which makes it ideally suited to high-volume applications with memory restrictions or to applications where code density is an issue.

The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the ARM968E-S processor has two instruction sets:

- Standard 32-bit ARMv5TE set
- 16-bit Thumb set

The Thumb set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit controller using 16-bit registers. This is possible because Thumb code operates on the same 32-bit register set as ARM code.

Thumb code can provide up to 65 % of the code size of ARM, and 160 % of the performance of an equivalent ARM controller connected to a 16-bit memory system.

The ARM968E-S processor is described in detail in the ARM968E-S data sheet Ref. 2.

2.3 On-chip flash memory system

The LPC2917/19 includes a 512 kB or 768 kB flash memory system. This memory can be used for both code and data storage. Programming of the flash memory can be accomplished in several ways. It may be programmed in-system via a serial port (e.g., CAN).

2.4 On-chip static RAM

In addition to the two 16 kB TCMs the LPC2917/19 includes two static RAM memories: one of 32 kB and one of 16 kB. Both may be used for code and/or data storage.

3. Features

3.1 General

- ARM968E-S processor at 80 MHz maximum.
- AHB system bus at 80 MHz.
- On-chip memory:
    - Two Tightly Coupled Memories (TCM), 16 kB Instruction TCM (ITCM), 16 kB Data TCM (DTCM).
    - Two separate internal SRAM instances; 32 kB and 16 kB.
    - Up to 768 kB flash program memory.
- Two-channel CAN controller supporting Full-CAN and extensive message filtering.
- Two LIN master controllers with full hardware support for LIN communication.
- Two 550 UARTs with 16-byte TX and RX FIFO depths.
- Three full-duplex queued SPIs with four slave-select lines; 16 bits wide; 8 locations deep; TX FIFO and RX FIFO.
- Four 32-bit timers each containing four capture-and-compare registers linked to I/Os.
- Four 6-channel PWMs with capture and trap functionality.
- 32-bit watchdog with timer change protection, running on safe clock.
- Up to 108 general-purpose I/O pins with programmable pull-up, pull-down or bus keeper.
- Vectored Interrupt Controller (VIC) with 16 priority levels.
- Two 8-channel 10-bit ADCs provide a total of up to 16 analog inputs, with conversion times as low as 2.44 μs per channel. Each channel provides a compare function to minimize interrupts.
- Up to 24 level-sensitive external interrupt pins, including CAN and LIN wake-up features.
- External Static Memory Controller (SMC) with eight memory banks; up to 32-bit data bus; up to 24-bit address bus.
- Processor wake-up from power-down via external interrupt pins; CAN or LIN activity.
- Flexible Reset Generation Unit (RGU) able to control resets of individual modules.
- Flexible Clock Generation Unit (CGU) able to control clock frequency of individual modules:
    - On-chip very low-power ring oscillator; fixed frequency of 0.4 MHz; always on to provide a Safe_Clock source for system monitoring.
    - On-chip crystal oscillator with a recommended operating range from 10 MHz to 25 MHz - maximum PLL input 15 MHz.
    - On-chip PLL allows CPU operation up to a maximum CPU rate of 80 MHz.
    - Generation of up to 10 base clocks.
    - Seven fractional dividers.
- Highly configurable system Power Management Unit (PMU):
    - Clock control of individual modules.

  - Allows minimization of system operating power consumption in any configuration.
- Standard ARM test and debug interface with real-time in-circuit emulator.
- Boundary-scan test supported.
- Dual power supply:
  - CPU operating voltage: 1.8 V ± 5 %.
  - I/O operating voltage: 2.7 V to 3.6 V; inputs tolerant up to 5.5 V.
- 144-pin LQFP package.
- –40 °C to 85 °C ambient operating temperature range.

4. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
LPC2917FBD144	LQFP144	plastic low profile quad flat package; 144 leads; body 20 × 20 × 1.4 mm	SOT486-1
LPC2919FBD144	LQFP144	plastic low profile quad flat package; 144 leads; body 20 × 20 × 1.4 mm	SOT486-1

4.1 Ordering options

Table 2. Part options

Type number	Flash memory	RAM	SMC	LIN 2.0	Package
LPC2917FBD144	512 kB	80 kB (including TCMs)	32-bit	2	LQFP144
LPC2919FBD144	768 kB	80 kB (including TCMs)	32-bit	2	LQFP144

5. Block diagram

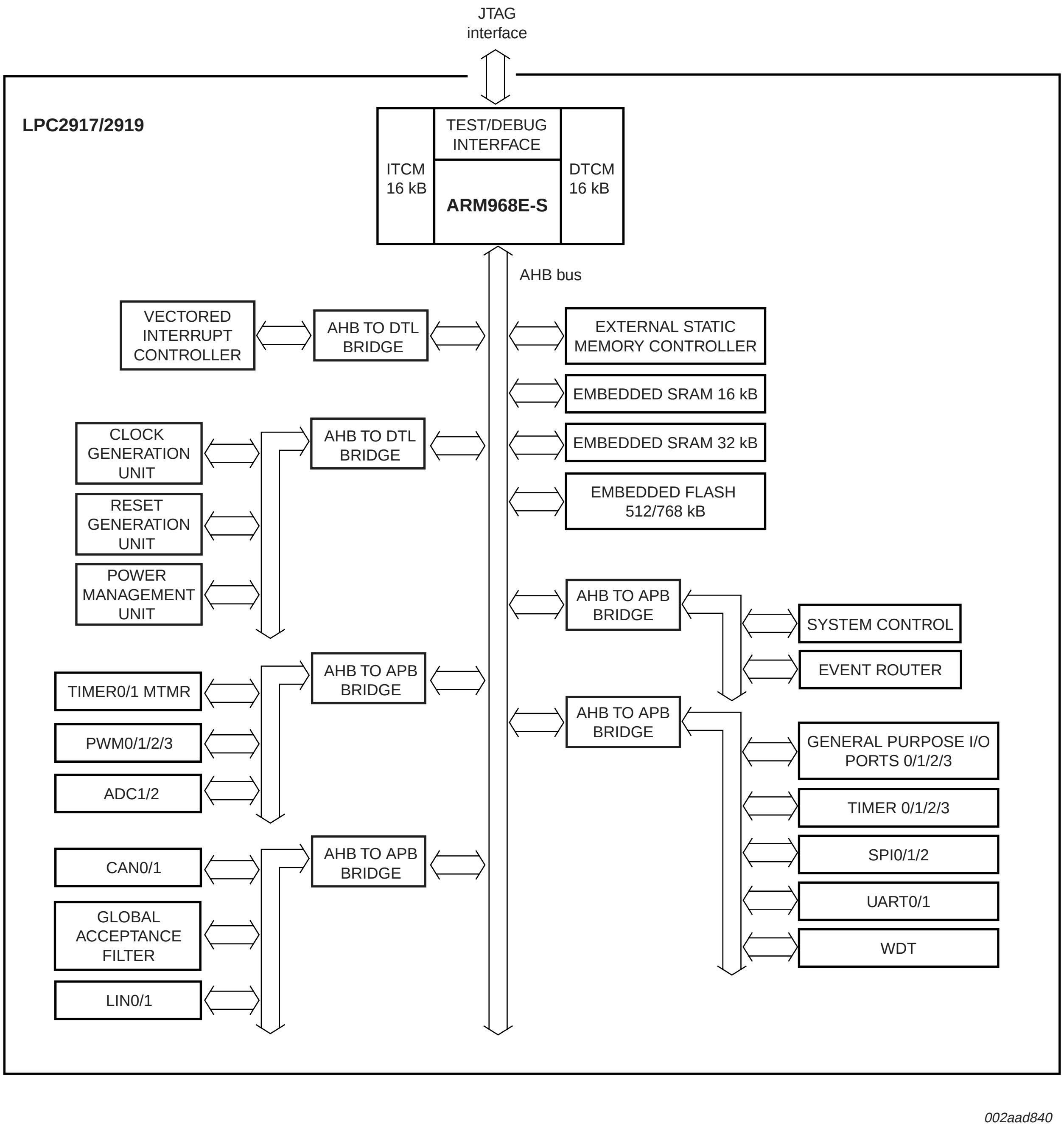


Fig 1. LPC2917/19 block diagram

6. Pinning information

6.1 Pinning

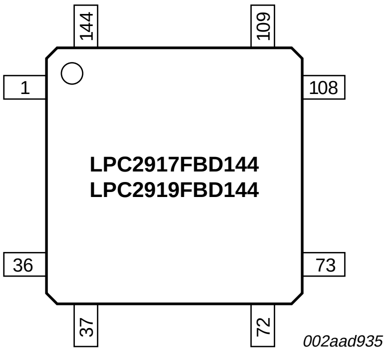


Fig 2. Pin configuration for SOT486-1 (LQFP144)

6.2 Pin description

6.2.1 General description

The LPC2917/19 has up to four ports: two of 32 pins each, one of 28 pins and one of 16 pins. The pin to which each function is assigned is controlled by the SFSP registers in the SCU. The functions combined on each port pin are shown in the pin description tables in this section.

6.2.2 LQFP144 pin assignment

Table 3. LQFP144 pin assignment

Pin name	Pin	Description			
		Default function	Function 1	Function 2	Function 3
TDO	1	IEEE 1149.1 test data out			
P2[21]/PCAP2[1]/D19	2	GPIO 2, pin 21	-	PWM2 CAP1	EXTBUS D19
P0[24]/TXD1/TXDC1/SCS2[0]	3	GPIO 0, pin 24	UART1 TXD	CAN1 TXDC	SPI2 SCS0
P0[25]/RXD1/RXDC1/SDO2	4	GPIO 0, pin 25	UART1 RXD	CAN1 RXDC	SPI2 SDO
P0[26]/SDI2	5	GPIO 0, pin 26	-	-	SPI2 SDI
P0[27]/SCK2	6	GPIO 0, pin 27	-	-	SPI2 SCK
P0[28]/CAP0[0]/MAT0[0]	7	GPIO 0, pin 28	-	TIMER0 CAP0	TIMER0 MAT0
P0[29]/CAP0[1]/MAT0[1]	8	GPIO 0, pin 29	-	TIMER0 CAP1	TIMER0 MAT1
$V_{DD(IO)}$	9	3.3 V power supply for I/O			
P2[22]/PCAP2[2]/D20	10	GPIO 2, pin 22	-	PWM2 CAP2	EXTBUS D20
P2[23]/PCAP3[0]/D21	11	GPIO 2, pin 23	-	PWM3 CAP0	EXTBUS D21
P3[6]/SCS0[3]/PMAT1[0]/TXDL1	12	GPIO 3, pin 6	SPI0 SCS3	PWM1 MAT0	LIN1 TXDL
P3[7]/SCS2[1]/PMAT1[1]/RXDL1	13	GPIO 3, pin 7	SPI2 SCS1	PWM1 MAT1	LIN1 RXDL
P0[30]/CAP0[2]/MAT0[2]	14	GPIO 0, pin 30	-	TIMER0 CAP2	TIMER0 MAT2
P0[31]/CAP0[3]/MAT0[3]	15	GPIO 0, pin 31	-	TIMER0 CAP3	TIMER0 MAT3

Table 3. LQFP144 pin assignment *…continued*

Pin name	Pin	Description			
		Default function	Function 1	Function 2	Function 3
P2[24]/PCAP3[1]/D22	16	GPIO 2, pin 24	-	PWM3 CAP1	EXTBUS D22
P2[25]/PCAP3[2]/D23	17	GPIO 2, pin 25	-	PWM3 CAP2	EXTBUS D23
$V_{DD(CORE)}$	18	1.8 V power supply for digital core			
$V_{SS(CORE)}$	19	ground for digital core			
P1[31]/CAP0[1]/MAT0[1]/EI5	20	GPIO 1, pin 31	TIMER0 CAP1	TIMER0 MAT1	EXTINT5
$V_{SS(IO)}$	21	ground for I/O			
P1[30]/CAP0[0]/MAT0[0]/EI4	22	GPIO 1, pin 30	TIMER0 CAP0	TIMER0 MAT0	EXTINT4
P3[8]/SCS2[0]/PMAT1[2]	23	GPIO 3, pin 8	SPI2 SCS0	PWM1 MAT2	-
P3[9]/SDO2/PMAT1[3]	24	GPIO 3, pin 9	SPI2 SDO	PWM1 MAT3	-
P1[29]/CAP1[0]/TRAP0/ PMAT3[5]	25	GPIO 1, pin 29	TIMER1 CAP0, EXT START	PWM TRAP0	PWM3 MAT5
P1[28]/CAP1[1]/TRAP1/ PMAT3[4]	26	GPIO 1, pin 28	TIMER1 CAP1, ADC1 EXT START	PWM TRAP1	PWM3 MAT4
P2[26]/CAP0[2]/MAT0[2]/EI6	27	GPIO 2, pin 26	TIMER0 CAP2	TIMER0 MAT2	EXTINT6
P2[27]/CAP0[3]/MAT0[3]/EI7	28	GPIO 2, pin 27	TIMER0 CAP3	TIMER0 MAT3	EXTINT7
P1[27]/CAP1[2]/TRAP2/ PMAT3[3]	29	GPIO 1, pin 27	TIMER1 CAP2, ADC2 EXT START	PWM TRAP2	PWM3 MAT3
P1[26]/PMAT2[0]/TRAP3/ PMAT3[2]	30	GPIO 1, pin 26	PWM2 MAT0	PWM TRAP3	PWM3 MAT2
$V_{DD(IO)}$	31	3.3 V power supply for I/O			
P1[25]/PMAT1[0]/PMAT3[1]	32	GPIO 1, pin 25	PWM1 MAT0	-	PWM3 MAT1
P1[24]/PMAT0[0]/PMAT3[0]	33	GPIO 1, pin 24	PWM0 MAT0	-	PWM3 MAT0
P1[23]/RXD0/CS5	34	GPIO 1, pin 23	UART0 RXD	-	EXTBUS CS5
P1[22]/TXD0/CS4	35	GPIO 1, pin 22	UART0 TXD	-	EXTBUS CS4
TMS	36	IEEE 1149.1 test mode select, pulled up internally			
TCK	37	IEEE 1149.1 test clock			
P1[21]/CAP3[3]/CAP1[3]/D7	38	GPIO 1, pin 21	TIMER3 CAP3	TIMER1 CAP3, MSCSS PAUSE	EXTBUS D7
P1[20]/CAP3[2]/SCS0[1]/D6	39	GPIO 1, pin 20	TIMER3 CAP2	SPI0 SCS1	EXTBUS D6
P1[19]/CAP3[1]/SCS0[2]/D5	40	GPIO 1, pin 19	TIMER3 CAP1	SPI0 SCS2	EXTBUS D5
P1[18]/CAP3[0]/SDO0/D4	41	GPIO 1, pin 18	TIMER3 CAP0	SPI0 SDO	EXTBUS D4
P1[17]/CAP2[3]/SDI0/D3	42	GPIO 1, pin 17	TIMER2 CAP3	SPI0 SDI	EXTBUS D3
$V_{SS(IO)}$	43	ground for I/O			
P1[16]/CAP2[2]/SCK0/D2	44	GPIO 1, pin 16	TIMER2 CAP2	SPI0 SCK	EXTBUS D2
P2[0]/MAT2[0]/TRAP3/D8	45	GPIO 2, pin 0	TIMER2 MAT0	PWM TRAP3	EXTBUS D8
P2[1]/MAT2[1]/TRAP2/D9	46	GPIO 2, pin 1	TIMER2 MAT1	PWM TRAP2	EXTBUS D9
P3[10]/SDI2/PMAT1[4]	47	GPIO 3, pin 10	SPI2 SDI	PWM1 MAT4	-
P3[11]/SCK2/PMAT1[5]	48	GPIO 3, pin 11	SPI2 SCK	PWM1 MAT5	-
P1[15]/CAP2[1]/SCS0[0]/D1	49	GPIO 1, pin 15	TIMER2 CAP1	SPI0 SCS0	EXTBUS D1
P1[14]/CAP2[0]/SCS0[3]/D0	50	GPIO 1, pin 14	TIMER2 CAP0	SPI0 SCS3	EXTBUS D0

Table 3. LQFP144 pin assignment *…continued*

Pin name	Pin	Description			
		Default function	**Function 1**	**Function 2**	**Function 3**
P1[13]/EI3/WE_N	51	GPIO 1, pin 13	EXTINT3	-	EXTBUS WE_N
P1[12]/EI2/OE_N	52	GPIO 1, pin 12	EXTINT2	-	EXTBUS OE_N
$V_{DD(IO)}$	53	3.3 V power supply for I/O			
P2[2]/MAT2[2]/TRAP1/D10	54	GPIO 2, pin 2	TIMER2 MAT2	PWM TRAP1	EXTBUS D10
P2[3]/MAT2[3]/TRAP0/D11	55	GPIO 2, pin 3	TIMER2 MAT3	PWM TRAP0	EXTBUS D11
P1[11]/SCK1/CS3	56	GPIO 1, pin 11	SPI1 SCK	-	EXTBUS CS3
P1[10]/SDI1/CS2	57	GPIO 1, pin 10	SPI1 SDI	-	EXTBUS CS2
P3[12]/SCS1[0]/EI4	58	GPIO 3, pin 12	SPI1 SCS0	EXTINT4	-
$V_{SS(CORE)}$	59	ground for digital core			
$V_{DD(CORE)}$	60	1.8 V power supply for digital core			
P3[13]/SDO1/EI5	61	GPIO 3, pin 13	SPI1 SDO	EXTINT5	-
P2[4]/MAT1[0]/EI0/D12	62	GPIO 2, pin 4	TIMER1 MAT0	EXTINT0	EXTBUS D12
P2[5]/MAT1[1]/EI1/D13	63	GPIO 2, pin 5	TIMER1 MAT1	EXTINT1	EXTBUS D13
P1[9]/SDO1/RXDL1/CS1	64	GPIO 1, pin 9	SPI1 SDO	LIN1 RXDL	EXTBUS CS1
$V_{SS(IO)}$	65	ground for I/O			
P1[8]/SCS1[0]/TXDL1/CS0	66	GPIO 1, pin 8	SPI1 SCS0	LIN1 TXDL	EXTBUS CS0
P1[7]/SCS1[3]/RXD1/A7	67	GPIO 1, pin 7	SPI1 SCS3	UART1 RXD	EXTBUS A7
P1[6]/SCS1[2]/TXD1/A6	68	GPIO 1, pin 6	SPI1 SCS2	UART1 TXD	EXTBUS A6
P2[6]/MAT1[2]/EI2/D14	69	GPIO 2, pin 6	TIMER1 MAT2	EXTINT2	EXTBUS D14
P1[5]/SCS1[1]/PMAT3[5]/A5	70	GPIO 1, pin 5	SPI1 SCS1	PWM3 MAT5	EXTBUS A5
P1[4]/SCS2[2]/PMAT3[4]/A4	71	GPIO 1, pin 4	SPI2 SCS2	PWM3 MAT4	EXTBUS A4
TRST_N	72	IEEE 1149.1 test reset NOT; active LOW; pulled up internally			
RST_N	73	asynchronous device reset; active LOW; pulled up internally			
$V_{SS(OSC)}$	74	ground for oscillator			
XOUT_OSC	75	crystal out for oscillator			
XIN_OSC	76	crystal in for oscillator			
$V_{DD(OSC)}$	77	1.8 V supply for oscillator			
$V_{SS(PLL)}$	78	ground for PLL			
P2[7]/MAT1[3]/EI3/D15	79	GPIO 2, pin 7	TIMER1 MAT3	EXTINT3	EXTBUS D15
P3[14]/SDI1/EI6/TXDC0	80	GPIO 3, pin 14	SPI1 SDI	EXTINT6	CAN0 TXDC
P3[15]/SCK1/EI7/RXDC0	81	GPIO 3, pin 15	SPI1 SCK	EXTINT7	CAN0 RXDC
$V_{DD(IO)}$	82	3.3 V power supply for I/O			
P2[8]/PMAT0[0]/SCS0[2]	83	GPIO 2, pin 8	-	PWM0 MAT0	SPI0 SCS2
P2[9]/PMAT0[1]/SCS0[1]	84	GPIO 2, pin 9	-	PWM0 MAT1	SPI0 SCS1
P1[3]/SCS2[1]/PMAT3[3]/A3	85	GPIO 1, pin 3	SPI2 SCS1	PWM3 MAT3	EXTBUS A3
P1[2]/SCS2[3]/PMAT3[2]/A2	86	GPIO 1, pin 2	SPI2 SCS3	PWM3 MAT2	EXTBUS A2
P1[1]/EI1/PMAT3[1]/A1	87	GPIO 1, pin 1	EXTINT1	PWM3 MAT1	EXTBUS A1
$V_{SS(CORE)}$	88	ground for digital core			
$V_{DD(CORE)}$	89	1.8 V power supply for digital core			
P1[0]/EI0/PMAT3[0]/A0	90	GPIO 1, pin 0	EXTINT0	PWM3 MAT0	EXTBUS A0

Table 3. LQFP144 pin assignment *…continued*

Pin name	Pin	Description			
		Default function	Function 1	Function 2	Function 3
P2[10]/PMAT0[2]/SCS0[0]	91	GPIO 2, pin 10	-	PWM0 MAT2	SPI0 SCS0
P2[11]/PMAT0[3]/SCK0	92	GPIO 2, pin 11	-	PWM0 MAT3	SPI0 SCK
P0[0]/TXDC0/D24	93	GPIO 0, pin 0	-	CAN0 TXDC	EXTBUS D24
$V_{SS(IO)}$	94	ground for I/O			
P0[1]/RXDC0/D25	95	GPIO 0, pin 1	-	CAN0 RXDC	EXTBUS D25
P0[2]/PMAT0[0]/D26	96	GPIO 0, pin 2	-	PWM0 MAT0	EXTBUS D26
P0[3]/PMAT0[1]/D27	97	GPIO 0, pin 3	-	PWM0 MAT1	EXTBUS D27
P3[0]/PMAT2[0]/CS6	98	GPIO 3, pin 0	-	PWM2 MAT0	EXTBUS CS6
P3[1]/PMAT2[1]/CS7	99	GPIO 3, pin 1	-	PWM2 MAT1	EXTBUS CS7
P2[12]/PMAT0[4]/SDI0	100	GPIO 2, pin 12	-	PWM0 MAT4	SPI0 SDI
P2[13]/PMAT0[5]/SDO0	101	GPIO 2, pin 13	-	PWM0 MAT5	SPI0 SDO
P0[4]/PMAT0[2]/D28	102	GPIO 0, pin 4	-	PWM0 MAT2	EXTBUS D28
P0[5]/PMAT0[3]/D29	103	GPIO 0, pin 5	-	PWM0 MAT3	EXTBUS D29
$V_{DD(IO)}$	104	3.3 V power supply for I/O			
P0[6]/PMAT0[4]/D30	105	GPIO 0, pin 6	-	PWM0 MAT4	EXTBUS D30
P0[7]/PMAT0[5]/D31	106	GPIO 0, pin 7	-	PWM0 MAT5	EXTBUS D31
$V_{DDA(ADC3V3)}$	107	3.3 V power supply for ADC			
JTAGSEL	108	TAP controller select input; LOW-level selects the ARM debug mode; HIGH-level selects boundary scan and flash programming; pulled up internally			
n.c.	109	not connected			
VREFP	110	HIGH reference for ADC			
VREFN	111	LOW reference for ADC			
P0[8]/IN1[0]/TXDL0/A20	112	GPIO 0, pin 8	ADC1 IN0	LIN0 TXDL	EXTBUS A20
P0[9]/IN1[1]/RXDL0/A21	113	GPIO 0, pin 9	ADC1 IN1	LIN0 RXDL	EXTBUS A21
P0[10]/IN1[2]/PMAT1[0]/A8	114	GPIO 0, pin 10	ADC1 IN2	PWM1 MAT0	EXTBUS A8
P0[11]/IN1[3]/PMAT1[1]/A9	115	GPIO 0, pin 11	ADC1 IN3	PWM1 MAT1	EXTBUS A9
P2[14]/PCAP0[0]/BLS0	116	GPIO 2, pin 14	-	PWM0 CAP0	EXTBUS BLS0
P2[15]/PCAP0[1]/BLS1	117	GPIO 2, pin 15	-	PWM0 CAP1	EXTBUS BLS1
P3[2]/MAT3[0]/PMAT2[2]	118	GPIO 3, pin 2	TIMER3 MAT0	PWM2 MAT2	-
$V_{SS(IO)}$	119	ground for I/O			
P3[3]/MAT3[1]/PMAT2[3]	120	GPIO 3, pin 3	TIMER3 MAT1	PWM2 MAT3	-
P0[12]/IN1[4]/PMAT1[2]/A10	121	GPIO 0, pin 12	ADC1 IN4	PWM1 MAT2	EXTBUS A10
P0[13]/IN1[5]/PMAT1[3]/A11	122	GPIO 0, pin 13	ADC1 IN5	PWM1 MAT3	EXTBUS A11
P0[14]/IN1[6]/PMAT1[4]/A12	123	GPIO 0, pin 14	ADC1 IN6	PWM1 MAT4	EXTBUS A12
P0[15]/IN1[7]/PMAT1[5]/A13	124	GPIO 0, pin 15	ADC1 IN7	PWM1 MAT5	EXTBUS A13
P0[16]IN2[0]/TXD0/A22	125	GPIO 0, pin 16	ADC2 IN0	UART0 TXD	EXTBUS A22
P0[17]/IN2[1]/RXD0/A23	126	GPIO 0, pin 17	ADC2 IN1	UART0 RXD	EXTBUS A23
$V_{DD(CORE)}$	127	1.8 V power supply for digital core			
$V_{SS(CORE)}$	128	ground for digital core			
P2[16]/TXD1/PCAP0[2]/BLS2	129	GPIO 2, pin 16	UART1 TXD	PWM0 CAP2	EXTBUS BLS2

Table 3. LQFP144 pin assignment *…continued*

Pin name	Pin	Description			
		Default function	Function 1	Function 2	Function 3
P2[17]/RXD1/PCAP1[0]/BLS3	130	GPIO 2, pin 17	UART1 RXD	PWM1 CAP0	EXTBUS BLS3
$V_{DD(IO)}$	131	3.3 V power supply for I/O			
P0[18]/IN2[2]/PMAT2[0]/A14	132	GPIO 0, pin 18	ADC2 IN2	PWM2 MAT0	EXTBUS A14
P0[19]/IN2[3]/PMAT2[1]/A15	133	GPIO 0, pin 19	ADC2 IN3	PWM2 MAT1	EXTBUS A15
P3[4]/MAT3[2]/PMAT2[4]/TXDC1	134	GPIO 3, pin 4	TIMER3 MAT2	PWM2 MAT4	CAN1 TXDC
P3[5]/MAT3[3]/PMAT2[5]/RXDC1	135	GPIO 3, pin 5	TIMER3 MAT3	PWM2 MAT5	CAN1 RXDC
P2[18]/PCAP1[1]/D16	136	GPIO 2, pin 18	-	PWM1 CAP1	EXTBUS D16
P2[19]/PCAP1[2]/D17	137	GPIO 2, pin 19	-	PWM1 CAP2	EXTBUS D17
P0[20]/IN2[4]/PMAT2[2]/A16	138	GPIO 0, pin 20	ADC2 IN4	PWM2 MAT2	EXTBUS A16
P0[21]/IN2[5]/PMAT2[3]/A17	139	GPIO 0, pin 21	ADC2 IN5	PWM2 MAT3	EXTBUS A17
P0[22]/IN2[6]/PMAT2[4]/A18	140	GPIO 0, pin 22	ADC2 IN6	PWM2 MAT4	EXTBUS A18
$V_{SS(IO)}$	141	ground for I/O			
P0[23]/IN2[7]/PMAT2[5]/A19	142	GPIO 0, pin 23	ADC2 IN7	PWM2 MAT5	EXTBUS A19
P2[20]/PCAP2[0]/D18	143	GPIO 2, pin 20	-	PWM2 CAP0	EXTBUS D18
TDI	144	IEEE 1149.1 data in, pulled up internally			

7. Functional description

7.1 Reset, debug, test and power description

7.1.1 Reset and power-up behavior

The LPC2917/19 contains external reset input and internal power-up reset circuits. This ensures that a reset is extended internally until the oscillators and flash have reached a stable state. See Section 11 for trip levels of the internal power-up reset circuit[1]. See Section 12 for characteristics of the several start-up and initialization times. Table 4 shows the reset pin.

Table 4. Reset pin

Symbol	Direction	Description
RST_N	IN	external reset input, active LOW; pulled up internally

At activation of the RST_N pin the JTAGSEL pin is sensed as logic LOW. If this is the case the LPC2917/19 is assumed to be connected to debug hardware, and internal circuits reprogram the source for the BASE_SYS_CLK to be the crystal oscillator instead of the Low-Power Ring Oscillator (LP_OSC). This is required because the clock rate when running at LP_OSC speed is too low for the external debugging environment.

7.1.2 Reset strategy

The LPC2917/19 contains a central module, the Reset Generation Unit (RGU) in the Power, Clock and Reset SubSystem (PCRSS), which controls all internal reset signals towards the peripheral modules. The RGU provides individual reset control as well as the monitoring functions needed for tracing a reset back to source.

---

1. Only for 1.8 V power sources

7.1.3 IEEE 1149.1 interface pins (JTAG boundary-scan test)

The LPC2917/19 contains boundary-scan test logic according to IEEE 1149.1, also referred to in this document as JTAG. The boundary-scan test pins can be used to connect a debugger probe for the embedded ARM processor. Pin JTAGSEL selects between boundary-scan mode and debug mode. Table 5 shows the boundary- scan test pins.

Table 5. IEEE 1149.1 boundary-scan test and debug interface

Symbol	Description
JTAGSEL	TAP controller select input. LOW-level selects ARM debug mode and HIGH-level selects boundary scan and flash programming; pulled up internally
TRST_N	test reset input; pulled up internally (active LOW)
TMS	test mode select input; pulled up internally
TDI	test data input, pulled up internally
TDO	test data output
TCK	test clock input

7.1.4 Power supply pins description

Table 6 shows the power supply pins.

Table 6. Power supplies

Symbol	Description
$V_{DD(CORE)}$	digital core supply 1.8 V
$V_{SS(CORE)}$	digital core ground (digital core, ADC1/2)
$V_{DD(IO)}$	I/O pins supply 3.3 V
$V_{SS(IO)}$	I/O pins ground
$V_{DD(OSC)}$	oscillator and PLL supply
$V_{SS(OSC)}$	oscillator ground
$V_{DDA(ADC3V3)}$	ADC1/2 3.3 V supply
$V_{SS(PLL)}$	PLL ground

7.2 Clocking strategy

7.2.1 Clock architecture

The LPC2917/19 contains several different internal clock areas. Peripherals like Timers, SPI, UART, CAN and LIN have their own individual clock sources called Base Clocks. All base clocks are generated by the Clock Generation Unit (CGU). They may be unrelated in frequency and phase and can have different clock sources within the CGU.

The system clock for the CPU and AHB Bus infrastructure has its own base clock. This means most peripherals are clocked independently from the system clock. See Figure 3 for an overview of the clock areas within the device.

Within each clock area there may be multiple branch clocks, which offers very flexible control for power-management purposes. All branch clocks are outputs of the Power Management Unit (PMU) and can be controlled independently. Branch clocks derived from the same base clock are synchronous in frequency and phase. See Section 8.8 for more details of clock and power control within the device.

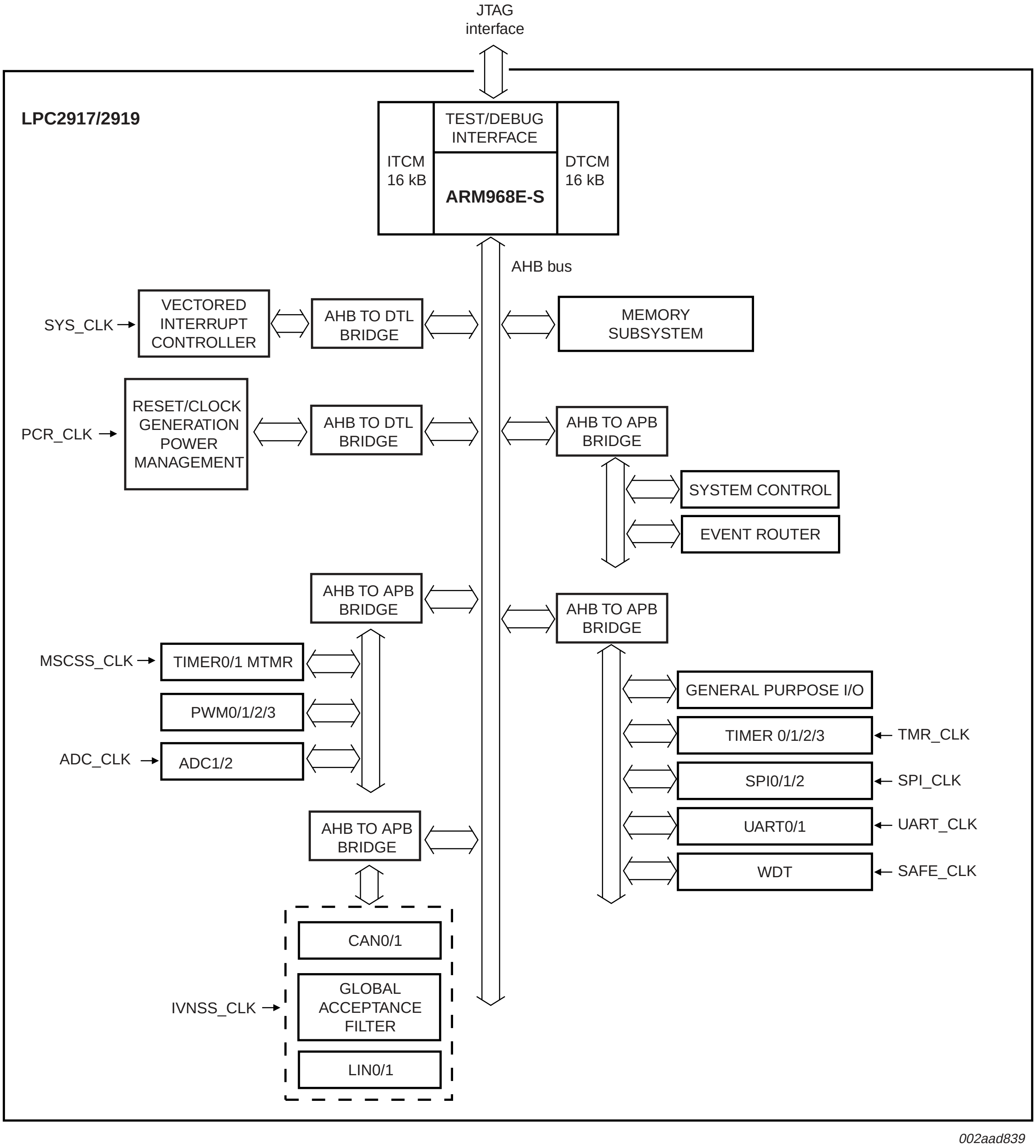


Fig 3. LPC2917/19 block diagram, overview of clock areas

7.2.2 Base clock and branch clock relationship

The next table contains an overview of all the base blocks in the LPC2917/19 and their derived branch clocks. A short description is given of the hardware parts that are clocked with the individual branch clocks. In relevant cases more detailed information can be found in the specific subsystem description. Some branch clocks have special protection since they clock vital system parts of the device and should (for example) not be switched off. See Section 8.8.6 for more details of how to control the individual branch clocks.

Table 7. Base clock and branch clock overview

Base clock	Branch clock name	Parts of the device clocked by this branch clock	Remark
BASE_SAFE_CLK	CLK_SAFE	watchdog timer	[1]
BASE_SYS_CLK	CLK_SYS_CPU	ARM968E-S and TCMs	
	CLK_SYS_SYS	AHB bus infrastructure	
	CLK_SYS_PCRSS	AHB side of bridge in PCRSS	
	CLK_SYS_FMC	Flash Memory Controller	
	CLK_SYS_RAM0	Embedded SRAM Controller 0 (32 kB)	
	CLK_SYS_RAM1	Embedded SRAM Controller 1 (16 kB)	
	CLK_SYS_SMC	External Static Memory Controller	
	CLK_SYS_GESS	General Subsystem	
	CLK_SYS_VIC	Vectored Interrupt Controller	
	CLK_SYS_PESS	Peripheral Subsystem	[2] [4]
	CLK_SYS_GPIO0	GPIO bank 0	
	CLK_SYS_GPIO1	GPIO bank 1	
	CLK_SYS_GPIO2	GPIO bank 2	
	CLK_SYS_GPIO3	GPIO bank 3	
	CLK_SYS_IVNSS_A	AHB side of bridge of IVNSS	
BASE_PCR_CLK	CLK_PCR_SLOW	PCRSS, CGU, RGU and PMU logic clock	[1], [3]
BASE_IVNSS_CLK	CLK_IVNSS_APB	APB side of the IVNSS	
	CLK_IVNSS_CANCA	CAN controller Acceptance Filter	
	CLK_IVNSS_CANC0	CAN channel 0	
	CLK_IVNSS_CANC1	CAN channel 1	
	CLK_IVNSS_LIN0	LIN channel 0	
	CLK_IVNSS_LIN1	LIN channel 1	
BASE_MSCSS_CLK	CLK_MSCSS_APB	APB side of the MSCSS	
	CLK_MSCSS_MTMR0	Timer 0 in the MSCSS	
	CLK_MSCSS_MTMR1	Timer 1 in the MSCSS	
	CLK_MSCSS_PWM0	PWM 0	
	CLK_MSCSS_PWM1	PWM 0	
	CLK_MSCSS_PWM2	PWM 0	
	CLK_MSCSS_PWM3	PWM 0	
	CLK_MSCSS_ADC1_APB	APB side of ADC 1	
	CLK_MSCSS_ADC2_APB	APB side of ADC 2	

Table 7. Base clock and branch clock overview *…continued*

Base clock	Branch clock name	Parts of the device clocked by this branch clock	Remark
BASE_UART_CLK	CLK_UART0	UART 0 interface clock	
	CLK_UART1	UART 1 interface clock	
BASE_SPI_CLK	CLK_SPI0	SPI 0 interface clock	
	CLK_SPI1	SPI 1 interface clock	
	CLK_SPI2	SPI 2 interface clock	
BASE_TMR_CLK	CLK_TMR0	Timer 0 clock for counter part	
	CLK_TMR1	Timer 1 clock for counter part	
	CLK_TMR2	Timer 2 clock for counter part	
	CLK_TMR3	Timer 3 clock for counter part	
BASE_ADC_CLK	CLK_ADC1	Control of ADC 1, capture sample result	
	CLK_ADC2	Control of ADC 2, capture sample result	
BASE_CLK_TESTSHELL	CLK_TESTSHELL_IP		

[1] This clock is always on (cannot be switched off for system safety reasons)

[2] In the peripheral subsystem parts of the Timers, watchdog timer, SPI and UART have their own clock source. See Section 8.4 for details.

[3] In the Power Clock and Reset Control subsystem parts of the CGU, RGU PMU have their own clock source. See Section 8.8 for details.

[4] The clock should remain activated when system wake-up on timer or UART is required.

8. Block description

8.1 Flash memory controller

8.1.1 Overview

The Flash Memory Controller (FMC) interfaces to the embedded flash memory for two tasks:

- Providing memory data transfer
- Memory configuration via triggering, programming and erasing

The flash memory has a 128-bit wide data interface and the flash controller offers two 128-bit buffer lines to improve system performance. The flash has to be programmed initially via JTAG. In-system programming must be supported by the bootloader. In-application programming is possible. Flash memory contents can be protected by disabling JTAG access. Suspension of burning or erasing is not supported.

The key features are:

- Programming by CPU via AHB
- Programming by external programmer via JTAG
- JTAG access protection
- Burn-finished and erase-finished interrupt

8.1.2 Description

After reset flash initialization is started, which takes t_{init} time, see Section 12. During this initialization flash access is not possible and AHB transfers to flash are stalled, blocking the AHB bus.

During flash initialization the index sector is read to identify the status of the JTAG access protection and sector security. If JTAG access protection is active the flash is not accessible via JTAG. ARM debug facilities are disabled to protect the flash memory contents against unwanted reading out externally. If sector security is active only the concerned sections are read.

Flash can be read synchronously or asynchronously to the system clock. In synchronous operation the flash goes into standby after returning the read data. Started reads cannot be stopped, and speculative reading and dual buffering are therefore not supported.

With asynchronous reading, transfer of the address to the flash and of read data from the flash is done asynchronously, giving the fastest possible response time. Started reads can be stopped, so speculative reading and dual buffering are supported.

Buffering is offered because the flash has a 128-bit wide data interface while the AHB interface has only 32 bits. With buffering a buffer line holds the complete 128-bit flash word, from which four words can be read. Without buffering every AHB data port read starts a flash read. A flash read is a slow process compared to the minimum AHB cycle time, so with buffering the average read time is reduced. This can improve system performance.

With single buffering the most recently read flash word remains available until the next flash read. When an AHB data-port read transfer requires data from the same flash word as the previous read transfer, no new flash read is done and the read data is given without wait cycles.

When an AHB data-port read transfer requires data from a different flash word to that involved in the previous read transfer, a new flash read is done and wait states are given until the new read data is available.

With dual buffering a secondary buffer line is used, the output of the flash being considered as the primary buffer. On a primary buffer hit data can be copied to the secondary buffer line, which allows the flash to start a speculative read of the next flash word.

Both buffer lines are invalidated after:

- Initialization
- Configuration-register access
- Data-latch reading
- Index-sector reading

The modes of operation are listed in Table 8.

Table 8. Flash read modes

Synchronous timing	
No buffer line	for single (non-linear) reads; one flash word read per word read
Single buffer line	default mode of operation; most recently read flash word is kept until another flash word is required
Asynchronous timing	
No buffer line	one flash word read per word read
Single buffer line	most recently read flash word is kept until another flash word is required
Dual buffer line, single speculative	on a buffer miss a flash read is done, followed by at most one speculative read; optimized for execution of code with small loops (less than eight words) from flash
Dual buffer line, always speculative	most recently used flash word is copied into second buffer line; next flash word read is started; highest performance for linear reads

8.1.3 Flash memory controller pin description

The flash memory controller has no external pins. However, the flash can be programmed via the JTAG pins, see Section 7.1.3.

8.1.4 Flash memory controller clock description

The flash memory controller is clocked by CLK_SYS_FMC, see Section 7.2.2.

8.1.5 Flash layout

The ARM processor can program the flash for ISP (In-System Programming) and IAP (In-Application Programming). Note that the flash always has to be programmed by ‘flash words’ of 128 bits (four 32-bit AHB bus words, hence 16 bytes).

The flash memory is organized into eight ‘small’ sectors of 8 kB each and up to 11 ‘large’ sectors of 64 kB each. The number of large sectors depends on the device type. A sector must be erased before data can be written to it. The flash memory also has sector-wise protection. Writing occurs per page which consists of 4096 bits (32 flash words). A small sector contains 16 pages; a large sector contains 128 pages.

Table 9 gives an overview of the flash sector base addresses.

Table 9. Flash sector overview

Sector number	Sector size (kB)	Sector base address
0	8	0000 0000h
1	8	0000 2000h
2	8	0000 4000h
3	8	0000 6000h
4	8	0000 8000h
5	8	0000 A000h
6	8	0000 C000h
7	8	0000 E000h
8	64	0001 0000h
9	64	0002 0000h
10	64	0003 0000h

Table 9. Flash sector overview *…continued*

Sector number	Sector size (kB)	Sector base address
11	64	0004 0000h
12	64	0005 0000h
13	64	0006 0000h
14	64	0007 0000h
15[1]	64	0008 0000h
16[1]	64	0009 0000h
17[1]	64	000A 0000h
18[1]	64	000B 0000h

[1] Availability of sector 15 to sector 18 depends on device type, see Section 4 “Ordering information”.

The index sector is a special sector in which the JTAG access protection and sector security are located. The address space becomes visible by setting the FS_ISS bit and overlaps the regular flash sector’s address space.

Note that the index sector cannot be erased, and that access to it has to be performed via code outside the flash.

8.1.6 Flash bridge wait-states

To eliminate the delay associated with synchronizing flash read data, a predefined number of wait-states must be programmed. These depend on flash memory response time and system clock period. The minimum wait-states value can be calculated with the following formulas:

Synchronous reading:

$$WST > \frac{t_{acc(clk)}}{t_{t_{tclk(sys)}}} - 1 \qquad (1)$$

Asynchronous reading:

$$WST > \frac{t_{acc(addr)}}{t_{tclk(sys)}} - 1 \qquad (2)$$

Remark: If the programmed number of wait-states is more than three, flash data reading cannot be performed at full speed (i.e., with zero wait-states at the AHB bus) if speculative reading is active.

8.2 External static memory controller

8.2.1 Overview

The LPC2917/19 contains an external Static Memory Controller (SMC) which provides an interface for external (off-chip) memory devices.

Key features are:

- Supports static memory-mapped devices including RAM, ROM, flash, burst ROM and external I/O devices

- Asynchronous page-mode read operation in non-clocked memory subsystems
- Asynchronous burst-mode read access to burst-mode ROM devices
- Independent configuration for up to eight banks, each up to 16 MB
- Programmable bus-turnaround (idle) cycles (one to 16)
- Programmable read and write wait states (up to 32), for static RAM devices
- Programmable initial and subsequent burst-read wait state for burst-ROM devices
- Programmable write protection
- Programmable burst-mode operation
- Programmable external data width: 8 bits, 16 bits or 32 bits
- Programmable read-byte lane enable control

8.2.2 Description

The SMC simultaneously supports up to eight independently configurable memory banks. Each memory bank can be 8 bits, 16 bits or 32 bits wide and is capable of supporting SRAM, ROM, burst-ROM memory or external I/O devices.

A separate chip select output is available for each bank. The chip select lines are configurable to be active HIGH or LOW. Memory-bank selection is controlled by memory addressing. Table 10 shows how the 32-bit system address is mapped to the external bus memory base addresses, chip selects and bank internal addresses.

Table 10. External memory-bank address bit description

32-bit system address bit field	Symbol	Description
31 to 29	BA[2:0]	external static-memory base address (three most significant bits); the base address can be found in the memory map; see Ref. 1. This field contains ‘010’ when addressing an external memory bank.
28 to 26	CS[2:0]	chip select address space for eight memory banks; see [1]
25 and 24	-	always ‘00’; other values are ‘mirrors’ of the 16 MB bank address
23 to 0	A[23:0]	16 MB memory banks address space

Table 11. External static-memory controller banks

CS[2:0]	Bank
000	bank 0
001	bank 1
010	bank 2
011	bank 3
100	bank 4
101	bank 5
110	bank 6
111	bank 7

8.2.3 External static-memory controller pin description

The external static-memory controller module in the LPC2917/19 has the following pins, which are combined with other functions on the port pins of the LPC2917/19. Table 12 shows the external memory controller pins.

Table 12. External memory controller pins

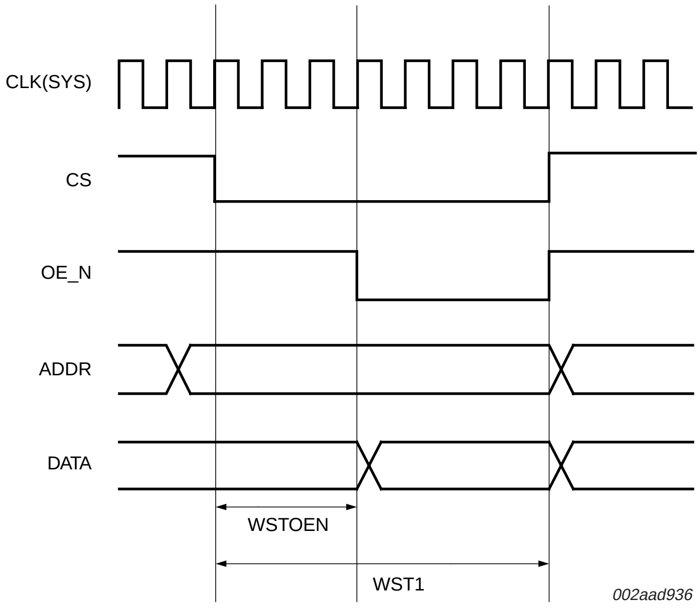
Symbol	Direction	Description
EXTBUS CSx	OUT	memory-bank x select, x runs from 0 to 7
EXTBUS BLSy	OUT	byte-lane select input y, y runs from 0 to 3
EXTBUS WE_N	OUT	write enable (active LOW)
EXTBUS OE_N	OUT	output enable (active LOW)
EXTBUS A[23:0]	OUT	address bus
EXTBUS D[31:0]	IN/OUT	data bus

8.2.4 External static-memory controller clock description

The External Static-Memory Controller is clocked by CLK_SYS_SMC, see Section 7.2.2.

8.2.5 External memory timing diagrams

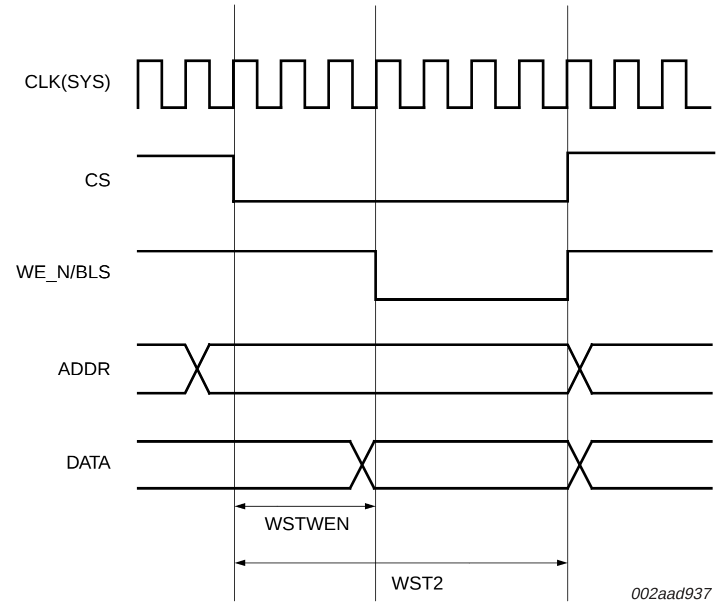
A timing diagram for reading from external memory is shown in Figure 4. The relationship between the wait-state settings is indicated with arrows.



WSTOEN = 3, WST1 = 7

Fig 4. Reading from external memory

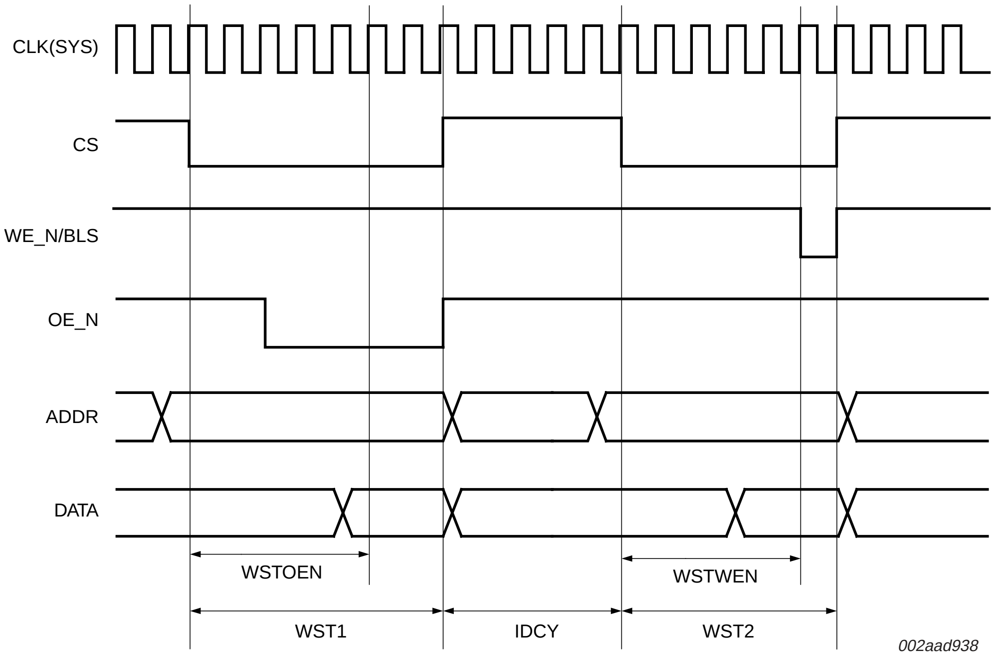
A timing diagram for writing to external memory is shown In Figure 5. The relationship between wait-state settings is indicated with arrows.



WSTWEN = 3, WST2 = 7

Fig 5. Writing to external memory

Usage of the idle/turn-around time (IDCY) is demonstrated In Figure 6. Extra wait states are added between a read and a write cycle in the same external memory device.



WSTOEN = 5, WSTWEN = 5, WST1 = 7, WST2 = 6, IDCY = 5

Fig 6. Reading/writing external memory

Address pins on the device are shared with other functions. When connecting external memories, check that the I/O pin is programmed for the correct function. Control of these settings is handled by the SCU.

8.3 General subsystem

8.3.1 General subsystem clock description

The general subsystem is clocked by CLK_SYS_GESS, see Section 7.2.2.

8.3.2 Chip and feature identification

8.3.2.1 Overview

The key features are:

- Identification of product
- Identification of features enabled

8.3.2.2 Description

The Chip/Feature ID (CFID) module contains registers which show and control the functionality of the chip. It contains an ID to identify the silicon, and also registers containing information about the features enabled or disabled on the chip.

8.3.2.3 CFID pin description

The CFID has no external pins.

8.3.3 System control unit

8.3.3.1 Overview

The SCU takes care of system-related functions.The key feature is configuration of the I/O port-pins multiplexer.

8.3.3.2 Description

The SCU defines the function of each I/O pin of the LPC2917/19. The I/O pin configuration should be consistent with peripheral function usage.

8.3.3.3 SCU pin description

The SCU has no external pins.

8.3.4 Event router

8.3.4.1 Overview

The event router provides bus-controlled routing of input events to the vectored interrupt controller for use as interrupt or wake-up signals.

Key features:

- Up to 24 level-sensitive external interrupt pins, including CAN, LIN and RXD wake-up features plus three internal event sources
- Input events can be used as interrupt source either directly or latched (edge-detected)
- Direct events disappear when the event becomes inactive

- Latched events remain active until they are explicitly cleared
- Programmable input level and edge polarity
- Event detection maskable
- Event detection is fully asynchronous, so no clock is required

8.3.4.2 Description

The event router allows the event source to be defined, its polarity and activation type to be selected and the interrupt to be masked or enabled. The event router can be used to start a clock on an external event.

The vectored interrupt-controller inputs are active HIGH.

8.3.4.3 Event-router pin description and mapping to register bit positions

The event router module in the LPC2917/19 is connected to the pins listed below. The pins are combined with other functions on the port pins of the LPC2917/19. Table 13 shows the pins connected to the event router, and also the corresponding bit position in the event-router registers and the default polarity.

Table 13. Event-router pin connections

Symbol	Direction	Bit position	Description	Default polarity
EXTINT0	IN	0	external interrupt input 0	1
EXTINT1	IN	1	external interrupt input 1	1
EXTINT2	IN	2	external interrupt input 2	1
EXTINT3	IN	3	external interrupt input 3	1
EXTINT4	IN	4	external interrupt input 4	1
EXTINT5	IN	5	external interrupt input 5	1
EXTINT6	IN	6	external interrupt input 6	1
EXTINT7	IN	7	external interrupt input 7	1
CAN0 RXDC	IN	8	CAN0 receive data input wake-up	0
CAN1 RXDC	IN	9	CAN1 receive data input wake-up	0
-	-	13 to 10	reserved	-
LIN0 RXDL	IN	14	LIN0 receive data input wake-up	0
LIN1 RXDL	IN	15	LIN1 receive data input wake-up	0
-	-	21 to 16	reserved	-
-	na	22	CAN interrupt (internal)	1
-	na	23	VIC FIQ (internal)	1
-	na	24	VIC IRQ (internal)	1
-	-	26 to 25	reserved	-

8.4 Peripheral subsystem

8.4.1 Peripheral subsystem clock description

The peripheral subsystem is clocked by a number of different clocks:

- CLK_SYS_PESS

- CLK_UART0/1
- CLK_SPI0/1/2
- CLK_TMR0/1/2/3
- CLK_SAFE see Section 7.2.2

8.4.2 Watchdog timer

8.4.2.1 Overview

The purpose of the watchdog timer is to reset the ARM9 processor within a reasonable amount of time if the processor enters an error state. The watchdog generates a system reset if the user program fails to trigger it correctly within a predetermined amount of time.

Key features:

- Internal chip reset if not periodically triggered
- Timer counter register runs on always-on safe clock
- Optional interrupt generation on watchdog time-out
- Debug mode with disabling of reset
- Watchdog control register change-protected with key
- Programmable 32-bit watchdog timer period with programmable 32-bit prescaler.

8.4.2.2 Description

The watchdog timer consists of a 32-bit counter with a 32-bit prescaler.

The watchdog should be programmed with a time-out value and then periodically restarted. When the watchdog times out it generates a reset through the RGU.

To generate watchdog interrupts in watchdog debug mode the interrupt has to be enabled via the interrupt enable register. A watchdog-overflow interrupt can be cleared by writing to the clear-interrupt register.

Another way to prevent resets during debug mode is via the Pause feature of the watchdog timer. The watchdog is stalled when the ARM9 is in debug mode and the PAUSE_ENABLE bit in the watchdog timer control register is set.

The Watchdog Reset output is fed to the Reset Generation Unit (RGU). The RGU contains a reset source register to identify the reset source when the device has gone through a reset. See Section 8.8.5.

8.4.2.3 Pin description

The watchdog has no external pins.

8.4.2.4 Watchdog timer clock description

The watchdog timer is clocked by two different clocks; CLK_SYS_PESS and CLK_SAFE, see Section 7.2.2. The register interface towards the system bus is clocked by CLK_SYS_PESS. The timer and prescale counters are clocked by CLK_SAFE which is always on.

8.4.3 Timer

8.4.3.1 Overview

The LPC2917/19 contains six identical timers: four in the peripheral subsystem and two in the Modulation and Sampling Control SubSystem (MSCSS) located at different peripheral base addresses. This section describes the four timers in the peripheral subsystem. Each timer has four capture inputs and/or match outputs. Connection to device pins depends on the configuration programmed into the port function-select registers. The two timers located in the MSCSS have no external capture or match pins, but the memory map is identical, see Section 8.7.7. One of these timers has an external input for a pause function.

The key features are:

- 32-bit timer/counter with programmable 32-bit prescaler
- Up to four 32-bit capture channels per timer. These take a snapshot of the timer value when an external signal connected to the TIMERx CAPn input changes state. A capture event may also optionally generate an interrupt
- Four 32-bit match registers per timer that allow:
  - Continuous operation with optional interrupt generation on match
  - Stop timer on match with optional interrupt generation
  - Reset timer on match with optional interrupt generation
- Up to four external outputs per timer corresponding to match registers, with the following capabilities:
  - Set LOW on match
  - Set HIGH on match
  - Toggle on match
  - Do nothing on match
- Pause input pin (MSCSS timers only)

8.4.3.2 Description

The timers are designed to count cycles of the clock and optionally generate interrupts or perform other actions at specified timer values, based on four match registers. They also include capture inputs to trap the timer value when an input signal changes state, optionally generating an interrupt. The core function of the timers consists of a 32 bit ‘prescale counter’ triggering the 32 bit ‘timer counter’. Both counters run on clock CLK_TMRx (x runs from 0 to 3) and all time references are related to the period of this clock. Note that each timer has its individual clock source within the Peripheral SubSystem. In the Modulation and Sampling SubSystem each timer also has its own individual clock source. See section Section 8.8.6 for information on generation of these clocks.

8.4.3.3 Pin description

The four timers in the peripheral subsystem of the LPC2917/19 have the pins described below. The two timers in the modulation and sampling subsystem have no external pins except for the pause pin on MSCSS timer 1. See Section 8.7.7 for a description of these timers and their associated pins. The timer pins are combined with other functions on the port pins of the LPC2917/19, see Section 8.3.3. Table Table 14 shows the timer pins (x runs from 0 to 3).

Table 14. Timer pins

Symbol	Direction	Description
TIMERx CAP[0]	IN	TIMER x capture input 0
TIMERx CAP[1]	IN	TIMER x capture input 1
TIMERx CAP[2]	IN	TIMER x capture input 2
TIMERx CAP[3]	IN	TIMER x capture input 3
TIMERx MAT[0]	OUT	TIMER x match output 0
TIMERx MAT[1]	OUT	TIMER x match output 1
TIMERx MAT[2]	OUT	TIMER x match output 2
TIMERx MAT[3]	OUT	TIMER x match output 3

8.4.3.4 Timer clock description

The timer modules are clocked by two different clocks; CLK_SYS_PESS and CLK_TMRx (x = 0-3), see Section 7.2.2. Note that each timer has its own CLK_TMRx branch clock for power management. The frequency of all these clocks is identical as they are derived from the same base clock BASE_CLK_TMR. The register interface towards the system bus is clocked by CLK_SYS_PESS. The timer and prescale counters are clocked by CLK_TMRx.

8.4.4 UARTs

8.4.4.1 Overview

The LPC2917/19 contains two identical UARTs located at different peripheral base addresses. The key features are:

- 16-byte receive and transmit FIFOs
- Register locations conform to 550 industry standard
- Receiver FIFO trigger points at 1 byte, 4 bytes, 8 bytes and 14 bytes
- Built-in baud rate generator

8.4.4.2 Description

The UART is commonly used to implement a serial interface such as RS232. The LPC2917/19 contains two industry-standard 550 UARTs with 16-byte transmit and receive FIFOs, but they can also be put into 450 mode without FIFOs.

8.4.4.3 UART pin description

The two UARTs in the LPC2917/19 have the following pins. The UART pins are combined with other functions on the port pins of the LPC2917/19. Table 15 shows the UART pins (x runs from 0 to 1).

Table 15. UART pins

Symbol	Direction	Description
UARTx TXD	OUT	UART channel x transmit data output
UARTx RXD	IN	UART channel x receive data input

8.4.4.4 UART clock description

The UART modules are clocked by two different clocks; CLK_SYS_PESS and CLK_UARTx (x = 0-1), see Section 7.2.2. Note that each UART has its own CLK_UARTx branch clock for power management. The frequency of all CLK_UARTx clocks is identical since they are derived from the same base clock BASE_CLK_UART. The register interface towards the system bus is clocked by CLK_SYS_PESS. The baud generator is clocked by the CLK_UARTx.

8.4.5 Serial peripheral interface

8.4.5.1 Overview

The LPC2917/19 contains three SPI modules to allow synchronous serial communication with slave or master peripherals.

The key features are:

- Master or slave operation
- Supports up to four slaves in sequential multi-slave operation
- Supports timer-triggered operation
- Programmable clock bit rate and prescale based on SPI source clock (BASE_SPI_CLK), independent of system clock
- Separate transmit and receive FIFO memory buffers; 16 bits wide, 32 locations deep
- Programmable choice of interface operation: Motorola SPI or Texas Instruments Synchronous Serial Interfaces
- Programmable data-frame size from 4 to 16 bits
- Independent masking of transmit FIFO, receive FIFO and receive overrun interrupts
- Serial clock-rate master mode: $fserial\_clk \leq f_{CLK(SPI)^*}/2$
- Serial clock-rate slave mode: $fserial\_clk = f_{CLK(SPI)^*}/4$
- Internal loopback test mode

8.4.5.2 Functional description

The SPI module is a master or slave interface for synchronous serial communication with peripheral devices that have either Motorola SPI or Texas Instruments Synchronous Serial Interfaces.

The SPI module performs serial-to-parallel conversion on data received from a peripheral device. The transmit and receive paths are buffered with FIFO memories (16 bits wide × 32 words deep). Serial data is transmitted on SPI_TXD and received on SPI_RXD.

The SPI module includes a programmable bit-rate clock divider and prescaler to generate the SPI serial clock from the input clock CLK_SPIx.

The SPI module’s operating mode, frame format, and word size are programmed through the SLVn_SETTINGS registers.

A single combined interrupt request SPI_INTREQ output is asserted if any of the interrupts are asserted and unmasked.

Depending on the operating mode selected, the SPI_CS_OUT outputs operate as an active-HIGH frame synchronization output for Texas Instruments synchronous serial frame format or an active-LOW chip select for SPI.

Each data frame is between four and 16 bits long, depending on the size of words programmed, and is transmitted starting with the MSB.

There are two basic frame types that can be selected:

- Texas Instruments synchronous serial
- Motorola Serial Peripheral Interface

8.4.5.3 Modes of operation

The SPI module can operate in:

- Master mode:
  - Normal transmission mode
  - Sequential slave mode
- Slave mode

8.4.5.4 SPI pin description

The three SPI modules in the LPC2917/19 have the pins listed below. The pins are combined with other functions on the port pins of the LPC2917/19, see Section 8.3.3. Table 16 shows the SPI pins (x runs from 0 to 2; y runs from 0 to 3).

Table 16. SPI pins

Symbol	Direction	Description
SPIx SCSy	IN/OUT	SPIx chip select[1][2]
SPIx SCK	IN/OUT	SPIx clock[1]
SPIx SDI	IN	SPIx data input
SPIx SDO	OUT	SPIx data output

[1] Direction of SPIx SCS and SPIx SCK pins depends on master or slave mode. These pins are output in master mode, input in slave mode.

[2] In slave mode there is only one chip select input pin, SPIx SCS0. The other chip selects have no function in slave mode.

8.4.5.5 SPI clock description

The SPI modules are clocked by two different clocks; CLK_SYS_PESS and CLK_SPIx (x = 0-2), see Section 7.2.2. Note that each SPI has its own CLK_SPIx branch clock for power management. The frequency of all clocks CLK_SPIx is identical as they are derived from the same base clock BASE_CLK_SPI. The register interface towards the system bus is clocked by CLK_SYS_PESS. The serial-clock rate divisor is clocked by CLK_SPIx.

The SPI clock frequency can be controlled by the CGU. In master mode the SPI clock frequency (CLK_SPIx) must be set to at least twice the SPI serial clock rate on the interface. In slave mode CLK_SPIx must be set to four times the SPI serial clock rate on the interface.

8.4.6 General-purpose I/O

8.4.6.1 Overview

The LPC2917/19 contains four general-purpose I/O ports located at different peripheral base addresses. In the 144-pin package all four ports are available. All I/O pins are bidirectional, and the direction can be programmed individually. The I/O pad behavior depends on the configuration programmed in the port function-select registers.

The key features are:

- General-purpose parallel inputs and outputs
- Direction control of individual bits
- Synchronized input sampling for stable input-data values
- All I/O defaults to input at reset to avoid any possible bus conflicts

8.4.6.2 Description

The general-purpose I/O provides individual control over each bidirectional port pin. There are two registers to control I/O direction and output level. The inputs are synchronized to achieve stable read-levels.

To generate an open-drain output, set the bit in the output register to the desired value. Use the direction register to control the signal. When set to output, the output driver actively drives the value on the output: when set to input the signal floats and can be pulled up internally or externally.

8.4.6.3 GPIO pin description

The five GPIO ports in the LPC2917/19 have the pins listed below. The GPIO pins are combined with other functions on the port pins of the LPC2917/19. Table 17 shows the GPIO pins.

Table 17. GPIO pins

Symbol	Direction	Description
GPIO0 pin[31:0]	IN/OUT	GPIO port x pins 31 to 0
GPIO1 pin[31:0]	IN/OUT	GPIO port x pins 31 to 0
GPIO2 pin[27:0]	IN/OUT	GPIO port x pins 27 to 0
GPIO3 pin[15:0]	IN/OUT	GPIO port x pins 15 to 0

8.4.6.4 GPIO clock description

The GPIO modules are clocked by several clocks, all of which are derived from BASE_SYS_CLK; CLK_SYS_PESS and CLK_SYS_GPIOx (x = 0-3), see Section 7.2.2. Note that each GPIO has its own CLK__SYS_GPIOx branch clock for power management. The frequency of all clocks CLK_SYS_GPIOx is identical to CLK_SYS_PESS since they are derived from the same base clock BASE_SYS_CLK.

8.5 CAN gateway

8.5.1 Overview

Controller Area Network (CAN) is the definition of a high-performance communication protocol for serial data communication. The two CAN controllers in the LPC2917/19 provide a full implementation of the CAN protocol according to the *CAN specification version 2.0B*. The gateway concept is fully scalable with the number of CAN controllers, and always operates together with a separate powerful and flexible hardware acceptance filter.

The key features are:

- Supports 11-bit as well as 29-bit identifiers
- Double receive buffer and triple transmit buffer
- Programmable error-warning limit and error counters with read/write access
- Arbitration-lost capture and error-code capture with detailed bit position
- Single-shot transmission (i.e., no re-transmission)
- Listen-only mode (no acknowledge; no active error flags)
- Reception of ‘own’ messages (self-reception request)
- Full CAN mode for message reception

8.5.2 Global acceptance filter

The global acceptance filter provides look-up of received identifiers - called acceptance filtering in CAN terminology - for all the CAN controllers. It includes a CAN ID look-up table memory, in which software maintains one to five sections of identifiers. The CAN ID look-up table memory is 2 kB large (512 words, each of 32 bits). It can contain up to 1024 standard frame identifiers or 512 extended frame identifiers or a mixture of both types. It is also possible to define identifier groups for standard and extended message formats.

8.5.3 CAN pin description

The two CAN controllers in the LPC2917/19 have the pins listed below. The CAN pins are combined with other functions on the port pins of the LPC2917/19. Table 18 shows the CAN pins (x runs from 0 to 1).

Table 18. CAN pins

Symbol	Direction	Description
CANx TXDC	OUT	CAN channel x transmit data output
CANx RXDC	IN	CAN channel x receive data input

8.6 LIN

8.6.1 Overview

The LPC2917/19 contain two LIN 2.0 master controllers. These can be used as dedicated LIN 2.0 master controllers with additional support for sync break generation and with hardware implementation of the LIN protocol according to spec 2.0.

The key features are:

- Complete LIN 2.0 message handling and transfer
- One interrupt per LIN message
- Slave response time-out detection
- Programmable sync-break length
- Automatic sync-field and sync-break generation
- Programmable inter-byte space
- Hardware or software parity generation
- Automatic checksum generation
- Fault confinement
- Fractional baud rate generator

8.6.2 LIN pin description

The two LIN 2.0 master controllers in the LPC2917/19 have the pins listed below. The LIN pins are combined with other functions on the port pins of the LPC2917/19. Table 19 shows the LIN pins. For more information see Ref. 1 subsection 3.43, LIN master controller.

Table 19. LIN controller pins

Symbol	Direction	Description
LIN0/1 TXDL	OUT	LIN channel 0/1 transmit data output
LIN0/1 RXDL	IN	LIN channel 0/1 receive data input

8.7 Modulation and sampling control subsystem

8.7.1 Overview

The Modulation and Sampling Control Subsystem (MSCSS) in the LPC2917/19 includes four Pulse-Width Modulators (PWMs), two 10-bit successive approximation Analog-to-Digital Converters (ADCs) and two timers.

The key features of the MSCSS are:

- Two 10-bit, 400 ksample/s, 8-channel ADCs with 3.3 V inputs and various trigger-start options
- Four 6-channel PWMs (Pulse-Width Modulators) with capture and trap functionality
- Two dedicated timers to schedule and synchronize the PWMs and ADCs

8.7.2 Description

The MSCSS contains Pulse-Width Modulators (PWMs), Analog-to-Digital Converters (ADCs) and timers.

Figure 7 provides an overview of the MSCSS. An AHB-to-APB bus bridge takes care of communication with the AHB system bus. Two internal timers are dedicated to this subsystem. MSCSS timer 0 can be used to generate start pulses for the ADCs and the first PWM. The second timer (MSCSS timer 1) is used to generate ‘carrier’ signals for the PWMs. These carrier patterns can be used, for example, in applications requiring current control. Several other trigger possibilities are provided for the ADCs (external, cascaded or following a PWM). The capture inputs of both timers can also be used to capture the start pulse of the ADCs.

The PWMs can be used to generate waveforms in which the frequency, duty cycle and rising and falling edges can be controlled very precisely. Capture inputs are provided to measure event phases compared to the main counter. Depending on the applications, these inputs can be connected to digital sensor motor outputs or digital external signals. Interrupt signals are generated on several events to closely interact with the CPU.

The ADCs can be used for any application needing accurate digitized data from analog sources. To support applications like motor control, a mechanism to synchronize several PWMs and ADCs is available (sync_in and sync_out).

Note that the PWMs run on the PWM clock and the ADCs on the ADC clock, see Section 8.8.4.

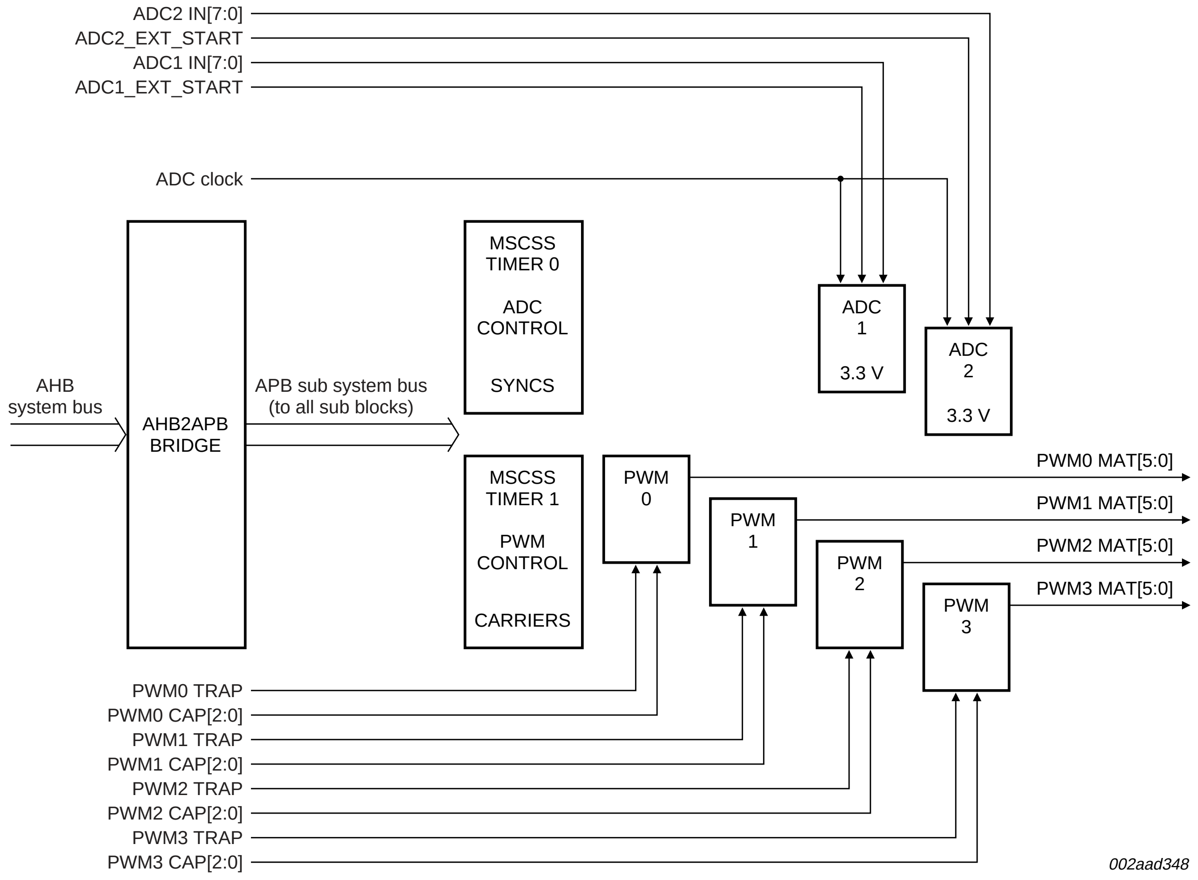


Fig 7. Modulation and sampling control subsystem block diagram

8.7.2.1 Synchronization and trigger features of the MSCSS

The MSCSS contains two internal timers to generate synchronization and carrier pulses for the ADCs and PWMs. Figure 8 shows how the timers are connected to the ADC and PWM modules.

Each ADC module has four start inputs. An ADC conversion is started when one of the start ADC conditions is valid:

- Start 0: ADC external start input pin; can be triggered at a positive or negative edge. Note that this signal is captured in the ADC clock domain
- Start 1: If the ‘preceding’ ADC conversion is ended, the sync_out signal starts an ADC conversion. This signal is captured in the MSCSS subsystem clock domain, see Section 8.7.5.2. As can be seen in Figure 8, the sync_out of ADC1 is connected to the start 1 input of ADC2 and the sync_out of ADC2 is connected to the start 1 input of ADC1.
- Start 2: The PWM sync_out can start an ADC conversion. The sync_out signal is synchronized to the ADC clock in the ADC module. This signal is captured in the MSCSS subsystem clock domain.
- Start 3: The match outputs from MSCSS timer 0 are connected to the start 3 inputs of the ADCs. This signal is captured in the ADC clock domain.

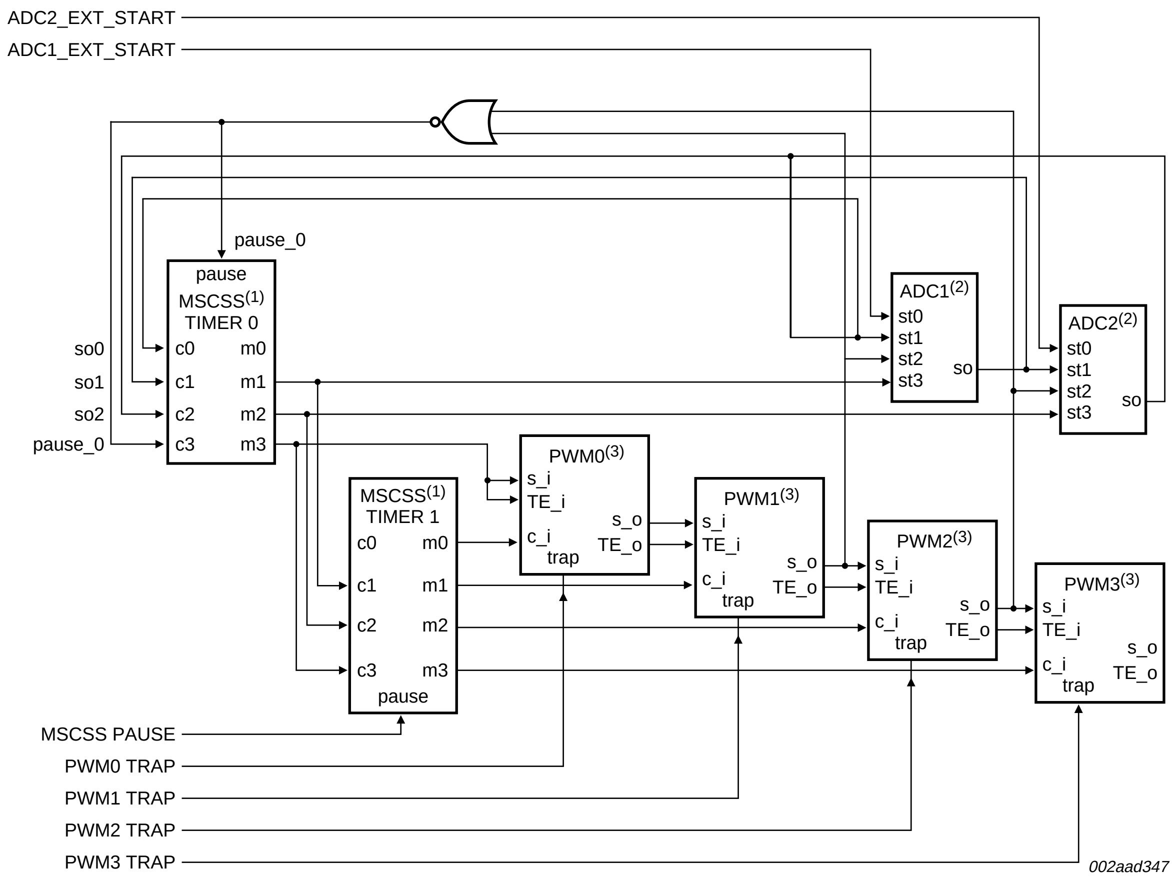
The PWM_sync and trans_enable_in of PWM 0 are connected to the 4th match output of MSCSS timer 0 to start the PWM after a pre-programmed delay. This sync signal is cascaded through all PWMs, allowing a programmable delay offset between subsequent PWMs. The sync delay of each PWM can be programmed synchronously or with a different phase for spreading the power load.

The match outputs of MSCSS timer 1 (PWM control) are connected to the corresponding carrier inputs of the PWM modules. The carrier signal is modulated with the PWM-generated waveforms.

The pause input of MSCSS timer 1 (PWM Control) is connected to an external input pin. Generation of the carrier signal is stopped by asserting the pause of this timer.

The pause input of MSCSS timer 0 (ADC Control) is connected to a ‘NOR’ of the PWM_sync outputs (start 2 input on the ADCs). If the pause feature of this timer is enabled the timer only counts when one of the PWM_sync outputs is active HIGH. This feature can be used to start the ADC once every x PWM cycles, where x corresponds to the value in the match register of the timer. In this case the start 3 input of the ADC should be enabled (start on match output of MSCSS timer 0).

The signals connected to the capture inputs of the timers (both MSCSS timer 0 and MSCSS timer 1) are intended for debugging.



(1) Timers:

c0 to c3 = capture in 0 to capture in 3

m0 to m3 = match out 0 to match out 3

(2) ADCs:

st0 to st3 = start 0 to start 3 inputs

s0 to s3 = sync_out 0 to sync_out 3

(3) PWMs:

c_i = carrier in

s_i = sync_in

s_o = sync_out

TE_i = trans_enable_in

TE_o = trans_enable_out

Fig 8. Modulation and sampling-control subsystem synchronization and triggering

8.7.3 MSCSS pin description

The pins of the LPC2917/19 MSCSS associated with the two ADC modules are described in Section 8.7.5.3. Pins directly connected to the four PWM modules are described in Section 8.7.6.5: pins directly connected to the MSCSS timer 1 module are described in Section 8.7.7.3.

8.7.4 MSCSS clock description

The MSCSS is clocked from a number of different sources:

- CLK_SYS_MSCSS_A clocks the AHB side of the AHB-to-APB bus bridge
- CLK_MSCSS_APB clocks the subsystem APB bus
- CLK_MSCSS_MTMR0/1 clocks the timers
- CLK_MSCSS_PWM0..3 clocks the PWMs.

Each ADC has two clock areas; a APB part clocked by CLK_MSCSS_ADCx_APB (x = 1 or 2) and a control part for the analog section clocked by CLK_ADCx = 1 or 2), see Section 7.2.2.

All clocks are derived from the BASE_MSCSS_CLK, except for CLK_SYS_MSCSS_A which is derived form BASE_SYS_CLK, and the CLK_ADCx clocks which are derived from BASE_CLK_ADC. If specific PWM or ADC modules are not used their corresponding clocks can be switched off.

8.7.5 Analog-to-digital converter

8.7.5.1 Overview

The MSCSS in the LPC2917/19 includes two 10-bit successive-approximation analog-to-digital converters.

The key features of the ADC interface module are:

- ADC1 and ADC2: Eight analog inputs; time-multiplexed; measurement range up to 3.3 V
- External reference-level inputs
- 400 ksample/s at 10-bit resolution up to 1500 ksample/s at 2-bit resolution
- Programmable resolution from 2-bit to 10-bit
- Single analog-to-digital conversion scan mode and continuous analog-to-digital conversion scan mode
- Optional conversion on transition on external start input, timer capture/match signal, PWM_sync or ‘previous’ ADC
- Converted digital values are stored in a register for each channel
- Optional compare condition to generate a ‘less than’ or an ‘equal to or greater than’ compare-value indication for each channel
- Power-down mode

8.7.5.2 Description

The ADC block diagram, Figure 9, shows the basic architecture of each ADC. The ADC functionality is divided into two major parts; one part running on the MSCSS Subsystem clock, the other on the ADC clock. This split into two clock domains affects the behavior from a system-level perspective. The actual analog-to-digital conversions take place in the ADC clock domain, but system control takes place in the system clock domain.

A mechanism is provided to modify configuration of the ADC and control the moment at which the updated configuration is transferred to the ADC domain.

The ADC clock is limited to 4.5 MHz maximum frequency and should always be lower than or equal to the system clock frequency. To meet this constraint or to select the desired lower sampling frequency the clock generation unit provides a programmable fractional system-clock divider dedicated to the ADC clock. Conversion rate is determined by the ADC clock frequency divided by the number of resolution bits plus one. Accessing ADC registers requires an enabled ADC clock, which is controllable via the clock generation unit, see Section 8.8.4.

Each ADC has four start inputs. Note that start 0 and start 2 are captured in the system clock domain while start 1 and start 3 are captured in the ADC domain. The start inputs are connected at MSCSS level, see Section 8.7.2.1 for details.

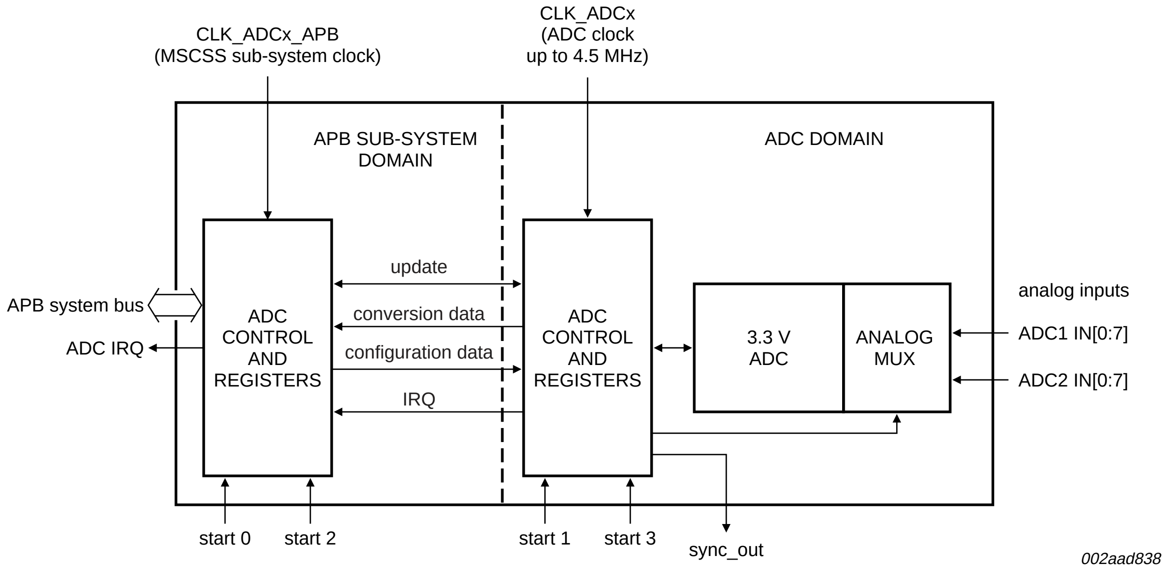


Fig 9. ADC block diagram

8.7.5.3 ADC pin description

The two ADC modules in the MSCSS have the pins described below. The ADCx input pins are combined with other functions on the port pins of the LPC2917/19. The VREFN and VREFP pins are common for both ADCs. Table 20 shows the ADC pins.

Table 20. Analog to digital converter pins

Symbol	Direction	Description
ADCn IN[7:0]	IN	analog input for ADCn, channel 7 to channel 0 (n is 1 or 2)
ADCn_EXT_START	IN	ADC external start-trigger input (n is 1 or 2)
VREFN	IN	ADC LOW reference level
VREFP	IN	ADC HIGH reference level

8.7.5.4 ADC clock description

The ADC modules are clocked from two different sources; CLK_MSCSS_ADCx_APB and CLK_ADCx (x = 1 or 2), see Section 7.2.2. Note that each ADC has its own CLK_ADCx and CLK_MSCSS_ADCx_APB branch clocks for power management. If an ADC is unused both its CLK_MSCSS_ADCx_APB and CLK_ADCx can be switched off.

The frequency of all the CLK_MSCSS_ADCx_APB clocks is identical to CLK_MSCSS_APB since they are derived from the same base clock BASE_MSCSS_CLK. Likewise the frequency of all the CLK_ADCx clocks is identical since they are derived from the same base clock BASE_ADC_CLK.

The register interface towards the system bus is clocked by CLK_MSCSS_ADCx_APB. Control logic for the analog section of the ADC is clocked by CLK_ADCx, see also Figure 9.

8.7.6 PWM

8.7.6.1 Overview

The MSCSS in the LPC2917/19 includes four PWM modules with the following features.

- Six pulse-width modulated output signals
- Double edge features (rising and falling edges programmed individually)
- Optional interrupt generation on match (each edge)
- Different operation modes: continuous or run-once
- 16-bit PWM counter and 16-bit prescale counter allow a large range of PWM periods
- A protective mode (TRAP) holding the output in a software-controllable state and with optional interrupt generation on a trap event
- Three capture registers and capture trigger pins with optional interrupt generation on a capture event
- Interrupt generation on match event, capture event, PWM counter overflow or trap event
- A burst mode mixing the external carrier signal with internally generated PWM
- Programmable sync-delay output to trigger other PWM modules (master/slave behavior)

8.7.6.2 Description

The ability to provide flexible waveforms allows PWM blocks to be used in multiple applications; e.g. automotive dimmer/lamp control and fan control. Pulse-width modulation is the preferred method for regulating power since no additional heat is generated and it is energy-efficient when compared with linear-regulating voltage control networks.

The PWM delivers the waveforms/pulses of the desired duty cycles and cycle periods. A very basic application of these pulses can be in controlling the amount of power transferred to a load. Since the duty cycle of the pulses can be controlled, the desired amount of power can be transferred for a controlled duration. Two examples of such applications are:

- Automotive dimmer controller: The flexibility of providing waves of a desired duty cycle and cycle period allows the PWM to control the amount of power to be transferred to the load. The PWM functions as a dimmer controller in this application
- Motor controller: The PWM provides multi-phase outputs, and these outputs can be controlled to have a certain pattern sequence. In this way the force/torque of the motor can be adjusted as desired. This makes the PWM function as a motor drive.

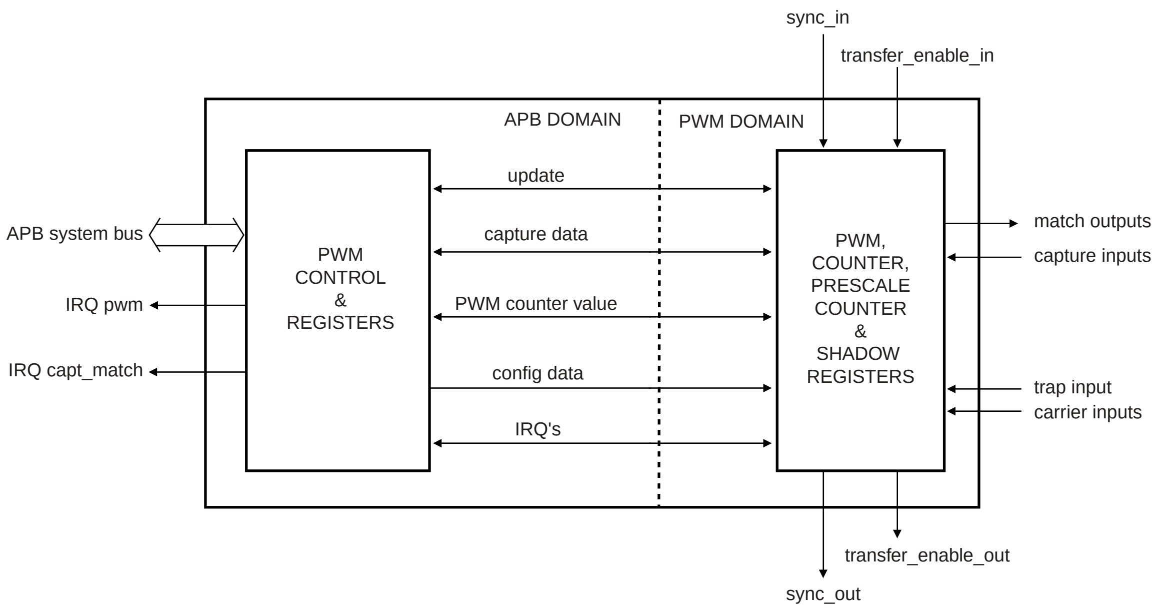


Fig 10. PWM block diagram

The PWM block diagram in Figure 10 shows the basic architecture of each PWM. PWM functionality is split into two major parts, a APB domain and a PWM domain, both of which run on clocks derived from the BASE_MSCSS_CLK. This split into two domains affects behavior from a system-level perspective. The actual PWM and prescale counters are located in the PWM domain but system control takes place in the APB domain.

The actual PWM consists of two counters; a 16-bit prescale counter and a 16-bit PWM counter. The position of the rising and falling edges of the PWM outputs can be programmed individually. The prescale counter allows high system bus frequencies to be scaled down to lower PWM periods. Registers are available to capture the PWM counter values on external events.

Note that in the Modulation and Sampling SubSystem, each PWM has its individual clock source CLK_MSCSS_PWMx (x runs from 0 to 3). Both the prescale and the timer counters within each PWM run on this clock CLK_MSCSS_PWMx, and all time references are related to the period of this clock. See Section 8.8 for information on generation of these clocks.

8.7.6.3 Synchronizing the PWM counters

A mechanism is included to synchronize the PWM period to other PWMs by providing a sync input and a sync output with programmable delay. Several PWMs can be synchronized using the trans_enable_in/trans_enable_out and sync_in/sync_out ports. See Section 8.7.2.1 for details of the connections of the PWM modules within the MSCSS in the LPC2917/19. PWM 0 can be master over PWM 1; PWM 1 can be master over PWM 2, etc.

8.7.6.4 Master and slave mode

A PWM module can provide synchronization signals to other modules (also called Master mode). The signal sync_out is a pulse of one clock cycle generated when the internal PWM counter (re)starts. The signal trans_enable_out is a pulse synchronous to sync_out, generated if a transfer from system registers to PWM shadow registers occurred when the PWM counter restarted. A delay may be inserted between the counter start and generation of trans_enable_out and sync_out.

A PWM module can use input signals trans_enable_in and sync_in to synchronize its internal PWM counter and the transfer of shadow registers (Slave mode).

8.7.6.5 PWM pin description

Each of the four PWM modules in the MSCSS has the following pins. These are combined with other functions on the port pins of the LPC2917/19. Table 21 shows the PWM0 to PWM3 pins.

Table 21. PWM pins

Symbol	Direction	Description
PWMn CAP[0]	IN	PWM n capture input 0
PWMn CAP[1]	IN	PWM n capture input 1
PWMn CAP[2]	IN	PWM n capture input 2
PWMn MAT[0]	OUT	PWM n match output 0
PWMn MAT[1]	OUT	PWM n match output 1
PWMn MAT[2]	OUT	PWM n match output 2
PWMn MAT[3]	OUT	PWM n match output 3
PWMn MAT[4]	OUT	PWM n match output 4
PWMn MAT[5]	OUT	PWM n match output 5
PWMn TRAP	IN	PWM n trap input

8.7.6.6 PWM clock description

The PWM modules are clocked by CLK_MSCSS_PWMx (x = 0-3), see Section 7.2.2. Note that each PWM has its own CLK_MSCSS_PWMx branch clock for power management. The frequency of all these clocks is identical to CLK_MSCSS_APB since they are derived from the same base clock BASE_MSCSS_CLK.

Also note that unlike the timer modules in the Peripheral SubSystem, the actual timer counter registers of the PWM modules run at the same clock as the APB system interface CLK_MSCSS_APB. This clock is independent of the AHB system clock.

If a PWM module is not used its CLK_MSCSS_PWMx branch clock can be switched off.

8.7.7 Timers in the MSCSS

8.7.7.1 Overview

The two timers in the MSCSS are functionally identical to the timers in the peripheral subsystem, see Section 8.4.3. The features of the timers in the MSCSS are the same as the timers in the peripheral subsystem, but the capture inputs and match outputs are not available on the device pins. These signals are instead connected to the ADC and PWM modules as outlined in the description of the MSCSS, see Section 8.7.2.

8.7.7.2 Description

See section Section 8.4.3.2 for a description of the timers.

8.7.7.3 MSCSS timer-pin description

MSCSS timer 0 has no external pins.

MSCSS timer 1 has a PAUSE pin available as external pin. The PAUSE pin is combined with other functions on the port pins of the LPC2917/19. Table 22 shows the MSCSS timer 1 external pin.

Table 22. MSCSS timer 1 pin

Symbol	Direction	Description
MSCSS PAUSE	IN	pause pin for MSCSS timer 1

8.7.7.4 MSCSS timer-clock description

The Timer modules in the MSCSS are clocked by CLK_MSCSS_MTMRx (x = 0-1), see Section 7.2.2. Note that each timer has its own CLK_MSCSS_MTMRx branch clock for power management. The frequency of all these clocks is identical to CLK_MSCSS_APB since they are derived from the same base clock BASE_MSCSS_CLK.

Note that, unlike the timer modules in the Peripheral SubSystem, the actual timer counter registers run at the same clock as the APB system interface CLK_MSCSS_APB. This clock is independent of the AHB system clock.

If a timer module is not used its CLK_MSCSS_MTMRx branch clock can be switched off.

8.8 Power, clock and reset control subsystem

8.8.1 Overview

The Power, Clock and Reset Control Subsystem (PCRSS) in the LPC2917/19 includes a Clock Generation Unit (CGU), a Reset Generation Unit (RGU) and a Power Management Unit (PMU).

8.8.2 Description

Figure 11 provides an overview of the PCRSS. An AHB-to-DTL bridge takes care of communication with the AHB system bus.

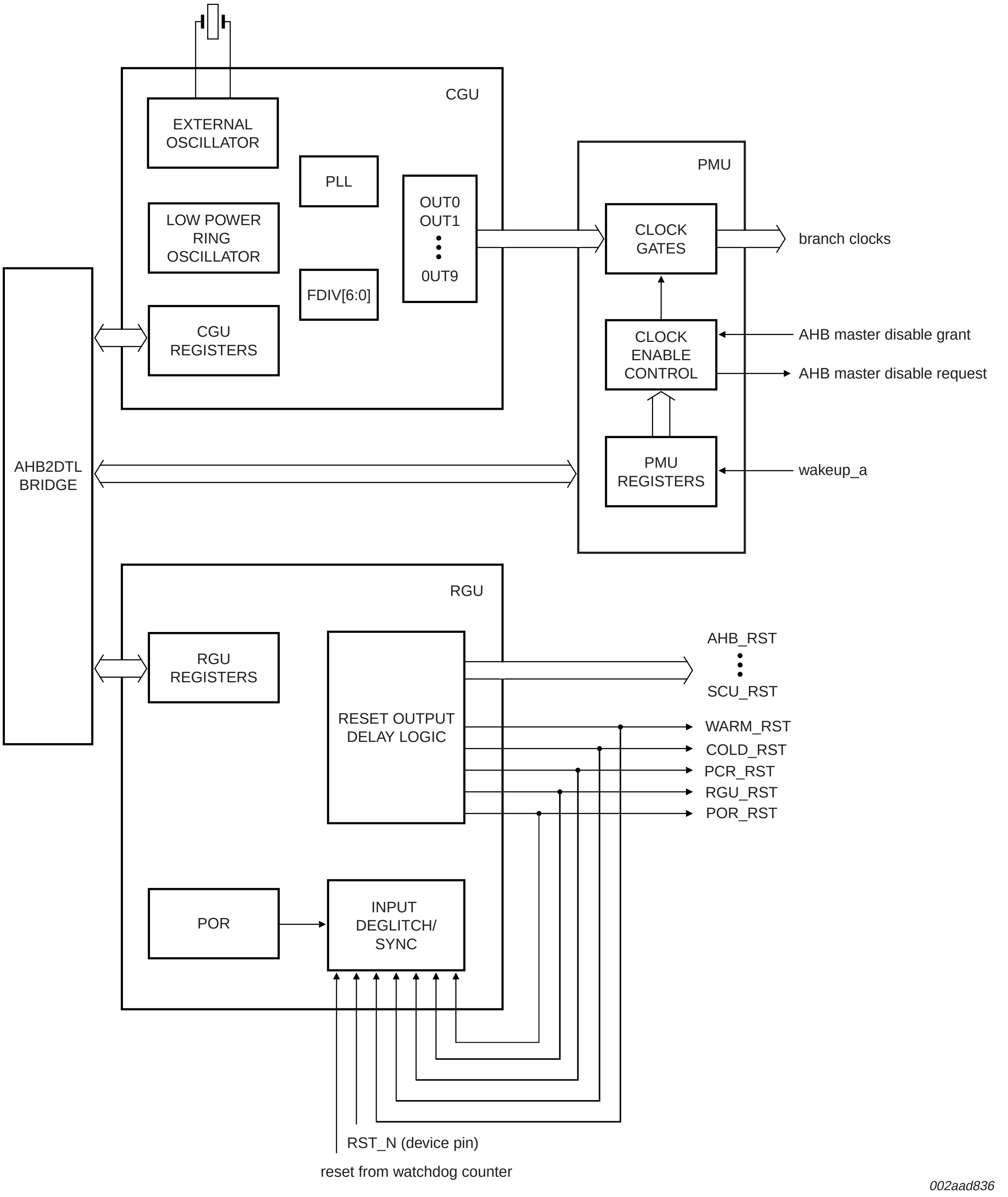


Fig 11. PCRSS block diagram

8.8.3 PCR subsystem clock description

The PCRSS is clocked by a number of different clocks. CLK_SYS_PCRSS clocks the AHB side of the AHB to DTL bus bridge and CLK_PCR_SLOW clocks the CGU, RGU and PMU internal logic, see Section 7.2.2. CLK_SYS_PCRSS is derived from BASE_SYS_CLK, which can be switched off in low-power modes. CLK_PCR_SLOW is derived from BASE_PCR_CLK and is always on in order to be able to wake up from low-power modes.

8.8.4 Clock Generation Unit (CGU)

8.8.4.1 Overview

The key features are:

- Generation of 10 and 2 test-base clocks, selectable from several embedded clock sources
- Crystal oscillator with power-down
- Control PLL with power-down
- Very low-power ring oscillator, always on to provide a ‘safe clock’
- Seven fractional clock dividers with L/D division
- Individual source selector for each base clock, with glitch-free switching
- Autonomous clock-activity detection on every clock source
- Protection against switching to invalid or inactive clock sources
- Embedded frequency counter
- Register write-protection mechanism to prevent unintentional alteration of clocks

Remark: Any clock-frequency adjustment has a direct impact on the timing of on-board peripherals such as the UARTs, SPI, watchdog, timers, CAN controller, LIN master controller, ADCs or flash memory interface.

8.8.4.2 Description

The clock generation unit provides 10 internal clock sources as described in Table 23.

Table 23. CGU base clocks

Number	Name	Frequency (MHz) [1]	Description
0	BASE_SAFE_CLK	0.4	base safe clock (always on)
1	BASE_SYS_CLK	80	base system clock
2	BASE_PCR_CLK	0.4 [2]	base PCR subsystem clock
3	BASE_IVNSS_CLK	80	base IVNSS subsystem clock
4	BASE_MSCSS_CLK	80	base MSCSS subsystem clock
5	BASE_UART_CLK	80	base UART clock
6	BASE_SPI_CLK	40	base SPI clock
7	BASE_TMR_CLK	80	base timers clock
8	BASE_ADC_CLK	4.5	base ADCs clock

[1] Maximum frequency that guarantees stable operation of the LPC2917/19.

[2] Fixed to low-power oscillator.

For generation of these base clocks, the CGU consists of primary and secondary clock generators and one output generator for each base clock.

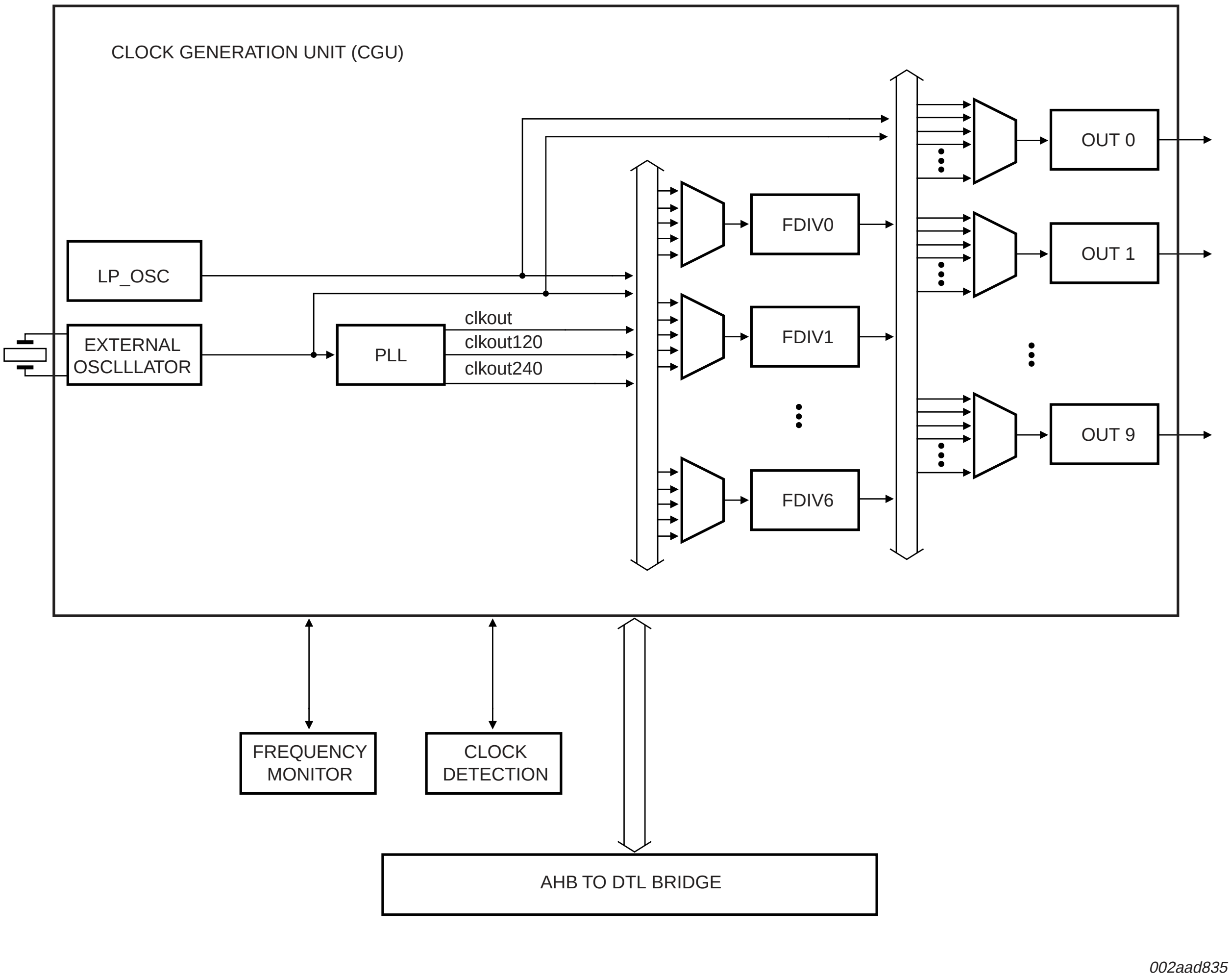


Fig 12. Block diagram of the CGU

There are two primary clock generators: a low-power ring oscillator (LP_OSC) and a crystal oscillator. See Figure 12.

LP_OSC is the source for the BASE_PCR_CLK that clocks the CGU itself and for BASE_SAFE_CLK that clocks a minimum of other logic in the device (like the watchdog timer). To prevent the device from losing its clock source LP_OSC cannot be put into power-down. The crystal oscillator can be used as source for high-frequency clocks or as an external clock input if a crystal is not connected.

Secondary clock generators are a PLL and seven fractional dividers (FDIV0..6). The PLL has three clock outputs: normal, 120° phase-shifted and 240° phase-shifted.

Configuration of the CGU: For every output generator - generating the base clocks - a choice can be made from the primary and secondary clock generators according to Figure 13.

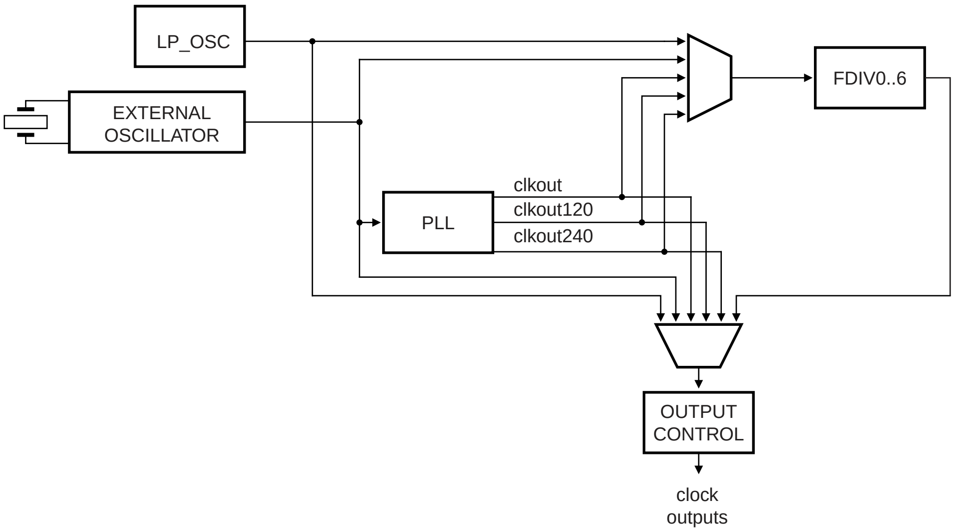


Fig 13. Structure of the clock generation scheme

Any output generator (except for BASE_SAFE_CLK and BASE_PCR_CLK) can be connected to either a fractional divider (FDIV0..6) or to one of the outputs of the PLL or to LP_OSC/crystal oscillator directly. BASE_SAFE_CLK and BASE_PCR_CLK can use only LP_OSC as source.

The fractional dividers can be connected to one of the outputs of the PLL or directly to LP_OSC/crystal Oscillator.

The PLL can be connected to the crystal oscillator.

In this way every output generating the base clocks can be configured to get the required clock. Multiple output generators can be connected to the same primary or secondary clock source, and multiple secondary clock sources can be connected to the same PLL output or primary clock source.

Invalid selections/programming - connecting the PLL to an FDIV or to one of the PLL outputs itself for example - will be blocked by hardware. The control register will not be written, the previous value will be kept, although all other fields will be written with new data. This prevents clocks being blocked by incorrect programming.

Default Clock Sources: Every secondary clock generator or output generator is connected to LP_OSC at reset. In this way the device runs at a low frequency after reset. It is recommended to switch BASE_SYS_CLK to a high-frequency clock generator as (one of) the first step(s) in the boot code after verifying that the high-frequency clock generator is running.

Clock Activity Detection: Clocks that are inactive are automatically regarded as invalid, and values of 'CLK_SEL' that would select those clocks are masked and not written to the control registers. This is accomplished by adding a clock detector to every clock generator. The RDET register keeps track of which clocks are active and inactive, and the appropriate ‘CLK_SEL’ values are masked and unmasked accordingly. Each clock detector can also generate interrupts at clock activation and deactivation so that the system can be notified of a change in internal clock status.

Clock detection is done using a counter running at the BASE_PCR_CLK frequency. If no positive clock edge occurs before the counter has 32 cycles of BASE_PCR_CLK the clock is assumed to be inactive. As BASE_PCR_CLK is slower than any of the clocks to be detected, normally only one BASE_PCR_CLK cycle is needed to detect activity. After reset all clocks are assumed to be ‘non-present’, so the RDET status register will be correct only after 32 BASE_PCR_CLK cycles.

Note that this mechanism cannot protect against a currently-selected clock going from active to inactive state. Therefore an inactive clock may still be sent to the system under special circumstances, although an interrupt can still be generated to notify the system.

Glitch-Free Switching: Provisions are included in the CGU to allow clocks to be switched glitch-free, both at the output generator stage and also at secondary source generators.

In the case of the PLL the clock will be stopped and held low for long enough to allow the PLL to stabilize and lock before being re-enabled. For all non-PLL Generators the switch will occur as quickly as possible, although there will always be a period when the clock is held low due to synchronization requirements.

If the current clock is high and does not go low within 32 cycles of BASE_PCR_CLK it is assumed to be inactive and is asynchronously forced low. This prevents deadlocks on the interface.

8.8.4.3 PLL functional description

A block diagram of the PLL is shown in Figure 14. The input clock is fed directly to the analog section. This block compares the phase and frequency of the inputs and generates the main clock[2]. These clocks are either divided by 2*P by the programmable post divider to create the output clock, or sent directly to the output. The main output clock is then divided by M by the programmable feedback divider to generate the feedback clock. The output signal of the analog section is also monitored by the lock detector to signal when the PLL has locked onto the input clock.

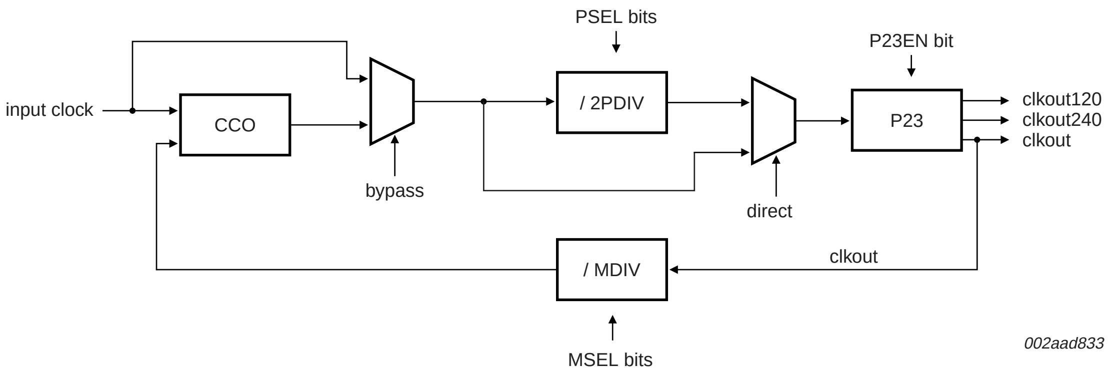


Fig 14. PLL block diagram

---

[2] Generation of the main clock is restricted by the frequency range of the PLL clock input. See Table 31, Dynamic characteristics.

Triple output phases: For applications that require multiple clock phases two additional clock outputs can be enabled by setting register P23EN to logic 1, thus giving three clocks with a 120° phase difference. In this mode all three clocks generated by the analog section are sent to the output dividers. When the PLL has not yet achieved lock the second and third phase output dividers run unsynchronized, which means that the phase relation of the output clocks is unknown. When the PLL LOCK register is set the second and third phase of the output dividers are synchronized to the main output clock CLKOUT PLL, thus giving three clocks with a 120° phase difference.

Direct output mode: In normal operating mode (with DIRECT set to logic 0) the CCO clock is divided by 2, 4, 8 or 16 depending on the value on the PSEL[1:0] input, giving an output clock with a 50 % duty cycle. If a higher output frequency is needed the CCO clock can be sent directly to the output by setting DIRECT to logic 1. Since the CCO does not directly generate a 50 % duty cycle clock, the output clock duty cycle in this mode can deviate from 50 %.

Power-down control: A Power-down mode has been incorporated to reduce power consumption when the PLL clock is not needed. This is enabled by setting the PD control register bit. In this mode the analog section of the PLL is turned off, the oscillator and the phase-frequency detector are stopped and the dividers enter a reset state. While in Power-down mode the LOCK output is low, indicating that the PLL is not in lock. When Power-down mode is terminated by clearing the PD control-register bit the PLL resumes normal operation, and makes the LOCK signal high once it has regained lock on the input clock.

8.8.4.4 CGU pin description

The CGU module in the LPC2917/19 has the pins listed in Table 24 below.

Table 24. CGU pins

Symbol	Direction	Description
XOUT_OSC	OUT	oscillator crystal output
XIN_OSC	IN	oscillator crystal input or external clock input

8.8.5 Reset Generation Unit (RGU)

8.8.5.1 Overview

The key features of the Reset Generation Unit (RGU) are:

- Reset controlled individually per subsystem
- Automatic reset stretching and release
- Monitor function to trace resets back to source
- Register write-protection mechanism to prevent unintentional resets

8.8.5.2 Description

The RGU controls all internal resets.

Each reset output is defined as a (combination of) reset input sources including the external reset input pins and internal power-on reset, see Table 25. The first five resets listed in this table form a sort of cascade to provide the multiple levels of impact that a reset may have. The combined input sources are logically OR-ed together so that activating any of the listed reset sources causes the output to go active.

Table 25. Reset output configuration

Reset output	Reset source	Parts of the device reset when activated
POR_RST	power-on reset module	LP_OSC; is source for RGU_RST
RGU_RST	POR_RST, RST_N pin	RGU internal; is source for PCR_RST
PCR_RST	RGU_RST, WATCHDOG	PCR internal; is source for COLD_RST
COLD_RST	PCR_RST	parts with COLD_RST as reset source below
WARM_RST	COLD_RST	parts with WARM_RST as reset source below
SCU_RST	COLD_RST	SCU
CFID_RST	COLD_RST	CFID
FMC_RST	COLD_RST	embedded Flash Memory Controller (FMC)
EMC_RST	COLD_RST	embedded SRAM Memory Controller
SMC_RST	COLD_RST	external Static Memory Controller (SMC)
GESS_A2V_RST	WARM_RST	GeSS AHB-to-APB bridge
PESS_A2V_RST	WARM_RST	PeSS AHB-to-APB bridge
GPIO_RST	WARM_RST	all GPIO modules
UART_RST	WARM_RST	all UART modules
TMR_RST	WARM_RST	all Timer modules in PeSS
SPI_RST	WARM_RST	all SPI modules
IVNSS_A2V_RST	WARM_RST	IVNSS AHB-to-APB bridge
IVNSS_CAN_RST	WARM_RST	all CAN modules including Acceptance filter
IVNSS_LIN_RST	WARM_RST	all LIN modules
MSCSS_A2V_RST	WARM_RST	MSCSS AHB to APB bridge
MSCSS_PWM_RST	WARM_RST	all PWM modules
MSCSS_ADC_RST	WARM_RST	all ADC modules
MSCSS_TMR_RST	WARM_RST	all Timer modules in MSCSS
VIC_RST	WARM_RST	Vectored Interrupt Controller (VIC)
AHB_RST	WARM_RST	CPU and AHB Bus infrastructure

8.8.5.3 RGU pin description

The RGU module in the LPC2917/19 has the following pins. Table 26 shows the RGU pins.

Table 26. RGU pins

Symbol	Direction	Description
RST_N	IN	external reset input, active LOW; pulled up internally

8.8.6 Power Management Unit (PMU)

8.8.6.1 Overview

This module enables software to actively control the system’s power consumption by disabling clocks not required in a particular operating mode.

Using the base clocks from the CGU as input, the PMU generates branch clocks to the rest of the LPC2917/19. Output clocks branched from the same base clock are phase- and frequency-related. These branch clocks can be individually controlled by software programming.

The key features are:

- Individual clock control for all LPC2917/19 sub-modules
- Activates sleeping clocks when a wake-up event is detected
- Clocks can be individually disabled by software
- Supports AHB master-disable protocol when AUTO mode is set
- Disables wake-up of enabled clocks when Power-down mode is set
- Activates wake-up of enabled clocks when a wake-up event is received
- Status register is available to indicate if an input base clock can be safely switched off (i.e., all branch clocks are disabled)

8.8.6.2 Description

The PMU controls all internal clocks of the device for power-mode management. With some exceptions, each branch clock can be switched on or off individually under control of software register bits located in its individual configuration register. Some branch clocks controlling vital parts of the device operate in a fixed mode. Table 27 shows which mode-control bits are supported by each branch clock.

By programming the configuration register the user can control which clocks are switched on or off, and which clocks are switched off when entering Power-down mode.

Note that the standby-wait-for-interrupt instructions of the ARM968E-S processor (putting the ARM CPU into a low-power state) are not supported. Instead putting the ARM CPU into power-down should be controlled by disabling the branch clock for the CPU.

Remark: For any disabled branch clocks to be re-activated their corresponding base clocks must be running (controlled by the CGU).

Table 27 shows the relation between branch and base clocks, see also Section 7.2.1. Every branch clock is related to one particular base clock: it is not possible to switch the source of a branch clock in the PMU.

Table 27. Branch clock overview

Legend:
‘1’ Indicates that the related register bit is tied off to logic HIGH, all writes are ignored
‘0’ Indicates that the related register bit is tied off to logic LOW, all writes are ignored
‘+’ Indicates that the related register bit is readable and writable

Branch clock name	Base clock	Implemented switch on/off mechanism		
		WAKE-UP	AUTO	RUN
CLK_SAFE	BASE_SAFE_CLK	0	0	1
CLK_SYS_CPU	BASE_SYS_CLK	+	+	1
CLK_SYS	BASE_SYS_CLK	+	+	1
CLK_SYS_PCR	BASE_SYS_CLK	+	+	1
CLK_SYS_FMC	BASE_SYS_CLK	+	+	+
CLK_SYS_RAM0	BASE_SYS_CLK	+	+	+
CLK_SYS_RAM1	BASE_SYS_CLK	+	+	+
CLK_SYS_SMC	BASE_SYS_CLK	+	+	+
CLK_SYS_GESS	BASE_SYS_CLK	+	+	+

Table 27. Branch clock overview *…continued*

Legend:
‘1’ Indicates that the related register bit is tied off to logic HIGH, all writes are ignored
‘0’ Indicates that the related register bit is tied off to logic LOW, all writes are ignored
‘+’ Indicates that the related register bit is readable and writable

Branch clock name	Base clock	Implemented switch on/off mechanism		
		WAKE-UP	AUTO	RUN
CLK_SYS_VIC	BASE_SYS_CLK	+	+	+
CLK_SYS_PESS	BASE_SYS_CLK	+	+	+
CLK_SYS_GPIO0	BASE_SYS_CLK	+	+	+
CLK_SYS_GPIO1	BASE_SYS_CLK	+	+	+
CLK_SYS_GPIO2	BASE_SYS_CLK	+	+	+
CLK_SYS_GPIO3	BASE_SYS_CLK	+	+	+
CLK_SYS_IVNSS_A	BASE_SYS_CLK	+	+	+
CLK_SYS_MSCSS_A	BASE_SYS_CLK	+	+	+
CLK_SYS_CHCA	BASE_SYS_CLK	+	+	+
CLK_SYS_CHCB	BASE_SYS_CLK	+	+	+
CLK_PCR_SLOW	BASE_PCR_CLK	+	+	1
CLK_IVNSS_APB	BASE_IVNSS_CLK	+	+	+
CLK_IVNSS_CANC0	BASE_IVNSS_CLK	+	+	+
CLK_IVNSS_CANC1	BASE_IVNSS_CLK	+	+	+
CLK_IVNSS_LIN0	BASE_IVNSS_CLK	+	+	+
CLK_IVNSS_LIN1	BASE_IVNSS_CLK	+	+	+
CLK_MSCSS_APB	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_MTMR0	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_MTMR1	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_PWM0	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_PWM1	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_PWM2	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_PWM3	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_ADC1_APB	BASE_MSCSS_CLK	+	+	+
CLK_MSCSS_ADC2_APB	BASE_MSCSS_CLK	+	+	+
CLK_UART0	BASE_UART_CLK	+	+	+
CLK_UART1	BASE_UART_CLK	+	+	+
CLK_SPI0	BASE_SPI_CLK	+	+	+
CLK_SPI1	BASE_SPI_CLK	+	+	+
CLK_SPI2	BASE_SPI_CLK	+	+	+
CLK_TMR0	BASE_TMR_CLK	+	+	+
CLK_TMR1	BASE_TMR_CLK	+	+	+
CLK_TMR2	BASE_TMR_CLK	+	+	+
CLK_TMR3	BASE_TMR_CLK	+	+	+

Table 27. Branch clock overview *…continued*

Legend:

'1' Indicates that the related register bit is tied off to logic HIGH, all writes are ignored

'0' Indicates that the related register bit is tied off to logic LOW, all writes are ignored

'+' Indicates that the related register bit is readable and writable

Branch clock name	Base clock	Implemented switch on/off mechanism		
		WAKE-UP	AUTO	RUN
CLK_ADC1	BASE_ADC_CLK	+	+	+
CLK_ADC2	BASE_ADC_CLK	+	+	+
CLK_TESTSHELL_IP	BASE_CLK_TESTSHELL	0	0	1

8.8.6.3 PMU pin description

The PMU has no external pins.

8.9 Vectored interrupt controller

8.9.1 Overview

The LPC2917/19 contains a very flexible and powerful Vectored Interrupt Controller (VIC) to interrupt the ARM processor on request.

The key features are:

- Level-active interrupt request with programmable polarity
- 56 interrupt-request inputs
- Software-interrupt request capability associated with each request input
- Observability of interrupt-request state before masking
- Software-programmable priority assignments to interrupt requests up to 15 levels
- Software-programmable routing of interrupt requests towards the ARM-processor inputs IRQ and FIQ
- Fast identification of interrupt requests through vector
- Support for nesting of interrupt service routines

8.9.2 Description

The Vectored Interrupt Controller routes incoming interrupt requests to the ARM processor. The interrupt target is configured for each interrupt request input of the VIC. The targets are defined as follows:

- Target 0 is ARM processor FIQ (fast interrupt service)
- Target 1 is ARM processor IRQ (standard interrupt service)

Interrupt-request masking is performed individually per interrupt target by comparing the priority level assigned to a specific interrupt request with a target-specific priority threshold. The priority levels are defined as follows:

- Priority level 0 corresponds to 'masked' (i.e., interrupt requests with priority 0 never lead to an interrupt)
- Priority 1 corresponds to the lowest priority

- Priority 15 corresponds to the highest priority

Software interrupt support is provided and can be supplied for:

- Testing Real-Time Operating System (RTOS) interrupt handling without using device-specific interrupt service routines
- Software emulation of an interrupt-requesting device, including interrupts

8.9.3 VIC pin description

The VIC module in the LPC2917/19 has no external pins.

8.9.4 VIC clock description

The VIC is clocked by CLK_SYS_VIC, see Section 7.2.2.

9. Limiting values

Table 28. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
Supply pins						
P_{tot}	total power dissipation		[1]	-	1	W
$V_{DD(CORE)}$	core supply voltage			–0.5	+2.0	V
$V_{DD(OSC\_PLL)}$	oscillator and PLL supply voltage			–0.5	+2.0	V
$V_{DDA(ADC3V3)}$	3.3 V ADC analog supply voltage			–0.5	+4.6	V
$V_{DD(IO)}$	I/O supply voltage			–0.5	+4.6	V
I_{DD}	supply current	average value per supply pin	[2]	-	98	mA
I_{SS}	ground current	average value per ground pin	[2]	-	98	mA
Input pins and I/O pins						
V_{XIN_OSC}	voltage on pin XIN_OSC			–0.5	+2.0	V
$V_{I(IO)}$	I/O input voltage		[3][4][5]	–0.5	$V_{DD(IO)}$ + 3.0	V
$V_{I(ADC)}$	ADC input voltage	I/O port 0.	[4][5]	–0.5	$V_{DDA(ADC3V3)}$ + 0.5	V
V_{VREFP}	voltage on pin VREFP			–0.5	+3.6	V
V_{VREFN}	voltage on pin VREFN			–0.5	+3.6	V
$I_{I(ADC)}$	ADC input current	average value per input pin	[2]	-	35	mA
Output pins and I/O pins configured as output						
I_{OHS}	HIGH-level short-circuit output current	drive HIGH, output shorted to $V_{SS(IO)}$	[9]	-	–33	mA
I_{OLS}	LOW-level short-circuit output current	drive LOW, output shorted to $_{VDD(IO)}$	[9]	-	+38	mA
General						
T_{stg}	storage temperature			–40	+150	°C
T_{amb}	ambient temperature			–40	+85	°C
T_{vj}	virtual junction temperature		[6]	–40	+125	°C
Memory						
$n_{endu(fl)}$	endurance of flash memory			-	100 000	cycle
$t_{ret(fl)}$	flash memory retention time			-	20	year

Table 28. Limiting values *…continued*

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
ESD						
V_{esd}	electrostatic discharge voltage	on all pins				
		human body model	[7]	–2000	+2000	V
		machine model	[8]	–200	+200	V
		charged device model		–500	+500	V
		on corner pins				
		charged device model		–750	+750	V

[1] Based on package heat transfer, not device power consumption.

[2] Peak current must be limited at 25 times average current.

[3] For I/O Port 0, the maximum input voltage is defined by $V_{I(ADC)}$.

[4] Only when $V_{DD(IO)}$ is present.

[5] Note that pull-up should be off. With pull-up do not exceed 3.6 V.

[6] In accordance with IEC 60747-1. An alternative definition of the virtual junction temperature is: $T_{vj} = T_{amb} + P_{tot} \times R_{th(j-a)}$ where $R_{th(j-a)}$ is a fixed value; see Section 10. The rating for T_{vj} limits the allowable combinations of power dissipation and ambient temperature.

[7] Human-body model: discharging a 100 pF capacitor via a 10 kΩ series resistor.

[8] Machine model: discharging a 200 pF capacitor via a 0.75 μH series inductance and 10 Ω resistor.

[9] 112 mA per $V_{DD(IO)}$ or $V_{SS(IO)}$ should not be exceeded.

10. Thermal characteristics

Table 29. Thermal characteristics

Symbol	Parameter	Conditions	Value	Unit
$R_{th(j-a)}$	thermal resistance from junction to ambient	in free air		
		package;		
		LQFP144	62	K/W

11. Static characteristics

Table 30. Static characteristics

$V_{DD(CORE)} = V_{DD(OSC\_PLL)}$; $V_{DD(IO)}$ = 2.7 V to 3.6 V; $V_{DDA(ADC3V3)}$ = 3.0 V to 3.6 V; T_{vj} = -40 °C to +125 °C; all voltages are measured with respect to ground; positive currents flow into the IC; unless otherwise specified.[1]

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Supplies							
Core supply							
$V_{DD(CORE)}$	core supply voltage			1.71	1.80	1.89	V
$I_{DD(CORE)}$	core supply current	ARM9 and all peripherals active at max clock speeds		-	1.1	2.5	mA/ MHz
		all clocks off	[2]	-	30	450	µA
I/O supply							
$V_{DD(IO)}$	I/O supply voltage			2.7	-	3.6	V
Oscillator supply							
$V_{DD(OSC\_PLL)}$	oscillator and PLL supply voltage			1.71	1.80	1.89	V
$I_{DD(OSC\_PLL)}$	oscillator and PLL supply current	start-up		3	-	4.5	mA
		normal mode		-	-	1	mA
		Power-down mode		-	-	2	µA
Analog-to-digital converter supply							
$V_{DDA(ADC3V3)}$	3.3 V ADC analog supply voltage			3.0	3.3	3.6	V
$I_{DDA(ADC3V3)}$	3.3 V ADC analog supply current	normal mode		-	-	1.9	mA
		Power-down mode		-	-	4	µA
Input pins and I/O pins configured as input							
V_I	input voltage	all port pins and $V_{DD(IO)}$ applied except port 0 pins 16 to 31 see Section 9	[7][8]	–0.5	-	+ 5.5	V
		port 0 pins 16 to 31	[8]			V_{VREFP}	
		all port pins and $V_{DD(IO)}$ not applied		–0.5	-	+3.6	V
		all other I/O pins, RESET_N, TRST_N, TDI, JTAGSEL, TMS, TCK		–0.5	-	$V_{DD(IO)}$	V
V_{IH}	HIGH-level input voltage	all port pins, RESET_N, TRST_N, TDI, JTAGSEL, TMS, TCK		2.0	-	-	V
V_{IL}	LOW-level input voltage	all port pins, RESET_N, TRST_N, TDI, JTAGSEL, TMS, TCK		-	-	0.8	V
V_{hys}	hysteresis voltage			0.4	-	-	V
I_{LIH}	HIGH-level input leakage current			-	-	1	µA

Table 30. Static characteristics *…continued*

$V_{DD(CORE)} = V_{DD(OSC\_PLL)}$; $V_{DD(IO)}$ = 2.7 V to 3.6 V; $V_{DDA(ADC3V3)}$ = 3.0 V to 3.6 V; T_{vj} = -40 °C to +125 °C; all voltages are measured with respect to ground; positive currents flow into the IC; unless otherwise specified.[1]

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I_{LIL}	LOW-level input leakage current			-	-	1	µA
$I_{I(pd)}$	pull-down input current	all port pins, V_I = 3.3 V; V_I = 5.5 V		25	50	100	µA
$I_{I(pu)}$	pull-up input current	all port pins, RESET_N, TRST_N, TDI, JTAGSEL, TMS: V_I = 0 V; V_I > 3.6 V is not allowed		–25	–50	–100	µA
C_i	input capacitance		[3]	-	3	8	pF
Output pins and I/O pins configured as output							
V_O	output voltage			0	-	$V_{DD(IO)}$	V
V_{OH}	HIGH-level output voltage	I_{OH} = –4 mA		$V_{DD(IO)}$ – 0.4	-	-	V
V_{OL}	LOW-level output voltage	I_{OL} = 4 mA		-	-	0.4	V
C_L	load capacitance			-	-	25	pF
Analog-to-digital converter supply							
V_{VREFN}	voltage on pin VREFN			0	-	V_{VREFP} – 2	V
V_{VREFP}	voltage on pin VREFP			V_{VREFN} + 2	-	$V_{DDA(ADC3V3)}$	V
$V_{I(ADC)}$	ADC input voltage	on port 0 pins		V_{VREFN}	-	V_{VREFP}	V
Z_i	input impedance	between V_{VREFN} and V_{VREFP}		4.4	-	-	kΩ
FSR	full scale range			2	-	10	bit
INL	integral non-linearity			–2	-	+2	LSB
DNL	differential non-linearity			–1	-	+1	LSB
$V_{err(offset)}$	offset error voltage			–20	-	+20	mV
$V_{err(FS)}$	full-scale error voltage			–20	-	+20	mV
Oscillator							
$R_{s(xtal)}$	crystal series resistance	f_{osc} = 10 MHz to 15 MHz	[5]				
		C_{xtal} = 10 pF; C_{ext} = 18 pF		-	-	160	Ω
		C_{xtal} = 20 pF; C_{ext} = 39 pF		-	-	60	Ω
		f_{osc} = 15 MHz to 20 MHz	[5]				
		C_{xtal} = 10 pF; C_{ext} = 18 pF		-	-	80	Ω
C_i	input capacitance	of XIN_OSC	[9]	-		2	pF

Table 30. Static characteristics *…continued*

$V_{DD(CORE)}$ = $V_{DD(OSC\_PLL)}$; $V_{DD(IO)}$ = 2.7 V to 3.6 V; $V_{DDA(ADC3V3)}$ = 3.0 V to 3.6 V; T_{vj} = -40 °C to +125 °C; all voltages are measured with respect to ground; positive currents flow into the IC; unless otherwise specified.[1]

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Power-up reset							
$V_{trip(high)}$	high trip level voltage		[6]	1.1	1.4	1.6	V
$V_{trip(low)}$	low trip level voltage		[6]	1.0	1.3	1.5	V
$V_{trip(dif)}$	difference between high and low trip level voltage		[6]	50	120	180	mV

[1] All parameters are guaranteed over the virtual junction temperature range by design. Pre-testing is performed at T_{amb} = 125 °C on wafer level. Cased products are tested at T_{amb} = 25 °C (final testing). Both pre-testing and final testing use correlated test conditions to cover the specified temperature and power-supply voltage range.

[2] Leakage current is exponential to temperature; worst-case value is at 125 °C T_{vj}. All clocks off. Analog modules and flash powered down.

[3] For Port 0, pin 0 to pin 15 add maximum 1.5 pF for input capacitance to ADC. For Port 0, pin 16 to pin 31 add maximum 1.0 pF for input capacitance to ADC.

[4] This value is the minimum drive capability. Maximum short-circuit output current is 33 mA (drive HIGH-level, shorted to ground) or –38 mA. (drive LOW-level, shorted to $V_{DD(IO)}$). The device will be damaged if multiple outputs are shorted.

[5] C_{xtal} is crystal load capacitance and C_{ext} are the two external load capacitors.

[6] The power-up reset has a time filter: $V_{DD(CORE)}$ must be above $V_{trip(high)}$ for 2 µs before reset is de-asserted; $V_{DD(CORE)}$ must be below $V_{trip(low)}$ for 11 µs before internal reset is asserted.

[7] Not 5 V-tolerant when pull-up is on.

[8] For I/O Port 0, the maximum input voltage is defined by $V_{I(ADC)}$.

[9] This parameter is not part of production testing or final testing, hence only a typical value is stated. Maximum and minimum values are based on simulation results.

12. Dynamic characteristics

Table 31. Dynamic characteristics

$V_{DD(CORE)} = V_{DD(OSC\_PLL)}$; $V_{DD(IO)}$ = 2.7 V to 3.6 V; $V_{DDA(ADC3V3)}$ = 3.0 V to 3.6 V; T_{vj} = −40 °C; all voltages are measured with respect to ground; positive currents flow into the IC; unless otherwise specified.[1]

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
I/O pins							
t_{THL}	HIGH-to-LOW transition time	C_L = 30 pF		4	-	13.8	ns
t_{TLH}	LOW-to-HIGH transition time	C_L = 30 pF		4	-	13.8	ns
Internal clock							
$f_{clk(sys)}$	system clock frequency		[2]	10	-	80	MHz
$T_{clk(sys)}$	system clock period		[2]	12.5	-	100	ns
Low-power ring oscillator							
$f_{ref(RO)}$	RO reference frequency			0.36	0.4	0.42	MHz
$t_{startup}$	start-up time	at maximum frequency	[3]	-	6	100	µs
Oscillator							
$f_{i(osc)}$	oscillator input frequency	maximum frequency is the clock input of an external clock source applied to the Xin pin		10	-	80	MHz
$t_{startup}$	start-up time	at maximum frequency	[3] [4]	-	500	-	µs
PLL							
$f_{i(PLL)}$	PLL input frequency			10	-	25	MHz
$f_{o(PLL)}$	PLL output frequency			10	-	160	MHz
		CCO; direct mode		156	-	320	MHz
Analog-to-digital converter							
$f_{i(ADC)}$	ADC input frequency		[5]	4	-	4.5	MHz
$f_{s(max)}$	maximum sampling rate	$f_{i(ADC)}$ = 4.5 MHz; $f_s = f_{i(ADC)}/(n + 1)$ with n = resolution					
		resolution 2 bit		-	-	1500	ksample/s
		resolution 10 bit		-	-	400	ksample/s
t_{conv}	conversion time	In number of ADC clock cycles		3	-	11	cycles
		In number of bits		2	-	10	bits
Flash memory							
t_{init}	initialization time			-	-	150	µs
$t_{wr(pg)}$	page write time			0.95	1	1.05	ms
$t_{er(sect)}$	sector erase time			95	100	105	ms
$t_{fl(BIST)}$	flash word BIST time			-	38	70	ns
$t_{a(clk)}$	clock access time			-	-	63.4	ns
$t_{a(A)}$	address access time			-	-	60.3	ns

Table 31. Dynamic characteristics *…continued*

$V_{DD(CORE)}$ = $V_{DD(OSC\_PLL)}$; $V_{DD(IO)}$ = 2.7 V to 3.6 V; $V_{DDA(ADC3V3)}$ = 3.0 V to 3.6 V; T_{vj} = –40 °C; all voltages are measured with respect to ground; positive currents flow into the IC; unless otherwise specified.[1]

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
External static memory controller							
$t_{a(R)int}$	internal read access time			-	-	20.5	ns
$t_{a(W)int}$	internal write access time			-	-	24.9	ns
UART							
f_{UART}	UART frequency			$\frac{1}{65024}f_{clk(uart)}$	-	$\frac{1}{2}f_{clk(uart)}$	MHz
SPI							
f_{SPI}	SPI operating frequency	master operation		$\frac{1}{65024}f_{clk(spi)}$	-	$\frac{1}{2}f_{clk(spi)}$	MHz
		slave operation		$\frac{1}{65024}f_{clk(spi)}$	-	$\frac{1}{4}f_{clk(spi)}$	MHz
Jitter specification							
$t_{jit(cc)(p-p)}$	cycle to cycle jitter (peak-to-peak value)	on CAN TXDC pin	[3]	-	0.4	1	ns

[1] All parameters are guaranteed over the virtual junction temperature range by design. Pre-testing is performed at T_{amb} = 125 °C ambient temperature on wafer level. Cased products are tested at T_{amb} = 25 °C (final testing). Both pre-testing and final testing use correlated test conditions to cover the specified temperature and power supply voltage range.

[2] See Table 23.

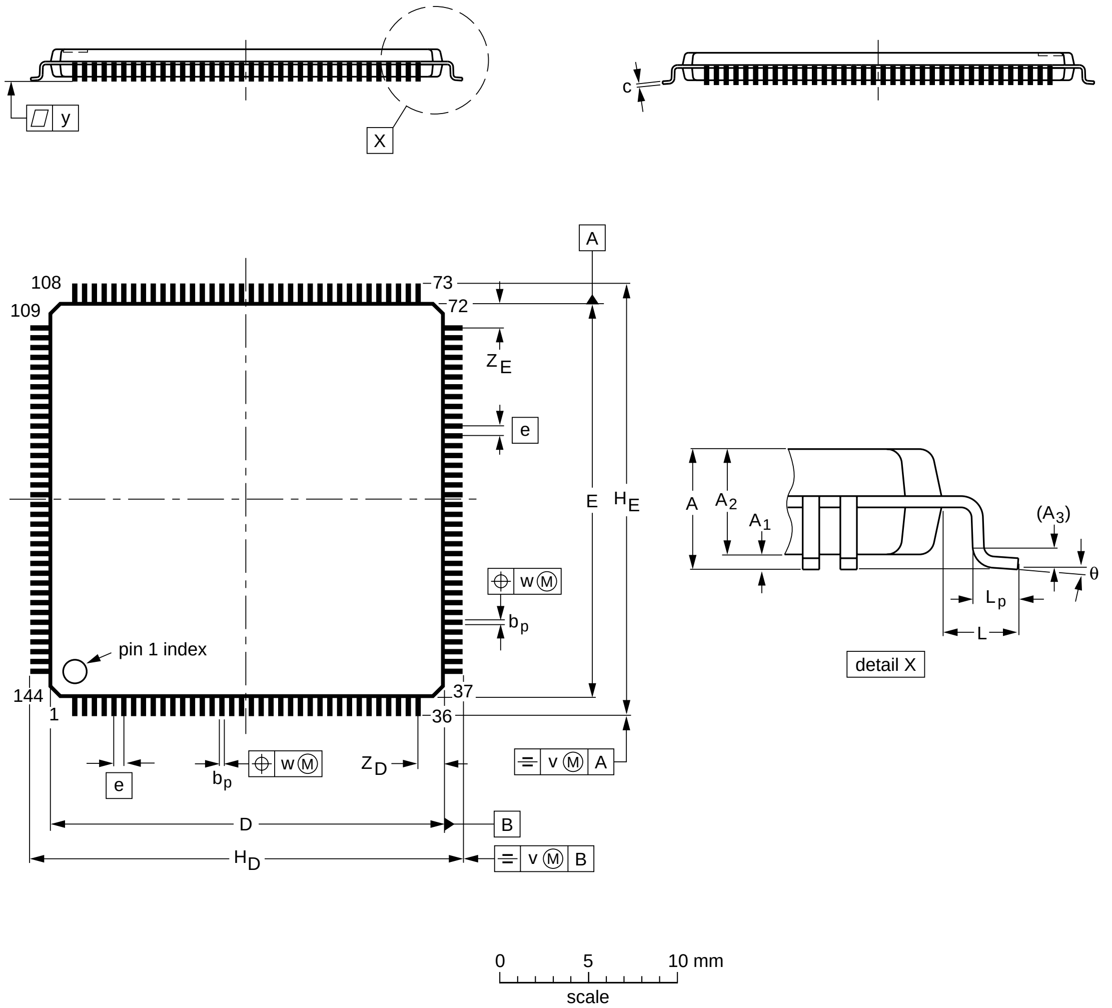
[3] This parameter is not part of production testing or final testing, hence only a typical value is stated.

[4] Oscillator start-up time depends on the quality of the crystal. For most crystals it takes about 1000 clock pulses until the clock is fully stable.

[5] Duty cycle clock should be as close as possible to 50 %.

13. Package outline

LQFP144: plastic low profile quad flat package; 144 leads; body 20 x 20 x 1.4 mm **SOT486-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	D(1)	E(1)	e	H_D	H_E	L	L_p	v	w	y	Z_D(1)	Z_E(1)	θ
mm	1.6	0.15 0.05	1.45 1.35	0.25	0.27 0.17	0.20 0.09	20.1 19.9	20.1 19.9	0.5	22.15 21.85	22.15 21.85	1	0.75 0.45	0.2	0.08	0.08	1.4 1.1	1.4 1.1	7° 0°

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT486-1	136E23	MS-026				~~00-03-14~~ 03-02-20

Fig 15. Package outline SOT486-1 (LQFP144)

14. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 “Surface mount reflow soldering description”*.

14.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

14.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

14.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

14.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 16) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 32 and 33

Table 32. SnPb eutectic process (from J-STD-020C)

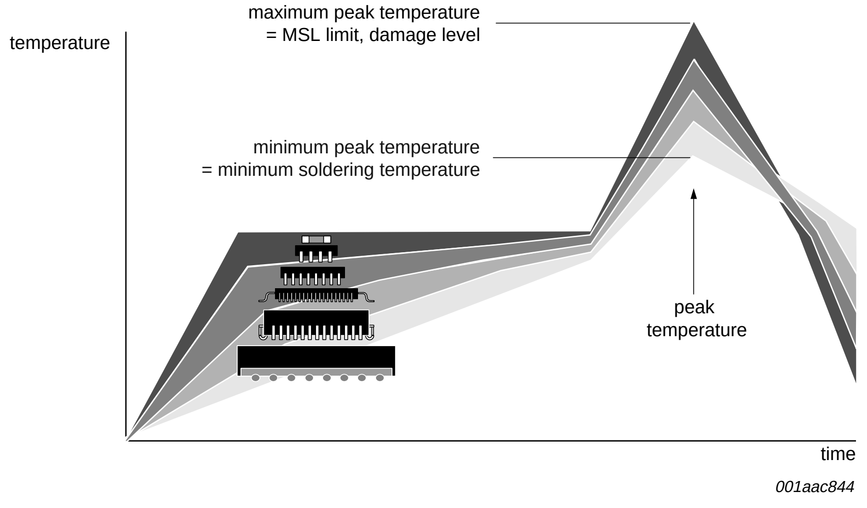
Package thickness (mm)	**Package reflow temperature (°C)**	
	Volume (mm^3)	
	< 350	**≥ 350**
< 2.5	235	220
≥ 2.5	220	220

Table 33. Lead-free process (from J-STD-020C)

Package thickness (mm)	**Package reflow temperature (°C)**		
	Volume (mm^3)		
	< 350	**350 to 2000**	**> 2000**
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 16.



MSL: Moisture Sensitivity Level

Fig 16. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 “Surface mount reflow soldering description”*.

15. Abbreviations

Table 34. Abbreviations list

Abbreviation	Description
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	ARM Peripheral Bus
BCL	Buffer Control List
BDL	Buffer Descriptor List
BEL	Buffer Entry List
BIST	Built-In Self Test
CAN	Controller Area Network
CCO	Current Controlled Oscillator
CISC	Complex Instruction Set Computers
DAC	Digital-to-Analog Converter
DTL	Device Transaction Level
FIFO	First In, First Out
FIQ	Fast Interrupt reQuest
GPIO	General Purpose Input/Output
I/O	Input/Output
IAP	In-Application Programming
IRQ	Interrupt ReQuest
ISP	In-System Programming
JTAG	Joint Test Action Group
LIN	Local Interconnect Network
MAC	Multiply-Accumulate
PLL	Phase-Locked Loop
PCRSS	Power, Clock and Reset SubSystem
PWM	Pulse Width Modulator
RISC	Reduced Instruction Set Computer
RTOS	Real-Time Operating System
RX	Receive
SFSP	SCU Function Select Port x,y (use without the P if there are no x,y)
SCL	Slot Control List
SCU	System Control Unit
SPI	Serial Peripheral Interface
SSP	Synchronous Serial Port
TAP	Test Access Port
TCM	Tightly Coupled Memory
TX	Transmit
UART	Universal Asynchronous Receiver Transmitter
VIC	Vectored Interrupt Controller

16. References

[1] **UM —** LPC2917/19 user manual

[2] **ARM —** ARM web site

[3] **ARM-SSP —** ARM primecell synchronous serial port (PL022) technical reference manual

[4] **CAN —** ISO 11898-1: 2002 road vehicles - Controller Area Network (CAN) - part 1: data link layer and physical signalling

[5] **LIN —** LIN specification package, revision 2.0

17. Revision history

Table 35. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
LPC2917_19_1	20080731	Product data sheet	-	-

18. Legal information

18.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term ‘short data sheet’ is explained in section “Definitions”.

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

18.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

18.3 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

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

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

18.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I^{2}C-bus — logo is a trademark of NXP B.V.

19. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

20. Contents

continued >>

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 31 July 2008
Document identifier: LPC2917_19_1