Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

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

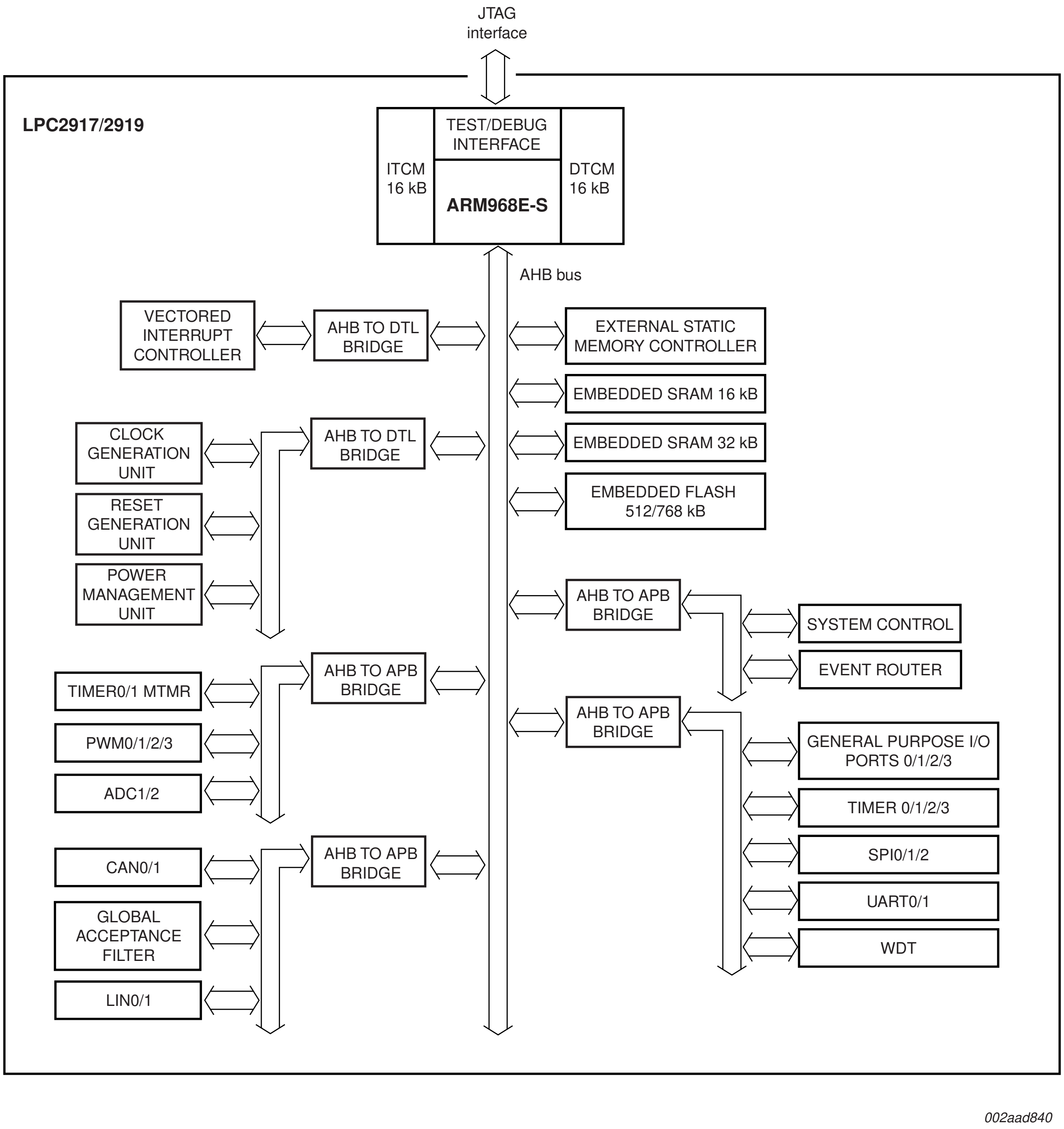


Fig 1. LPC2917/19 block diagram

6. Pinning information

6.1 Pinning

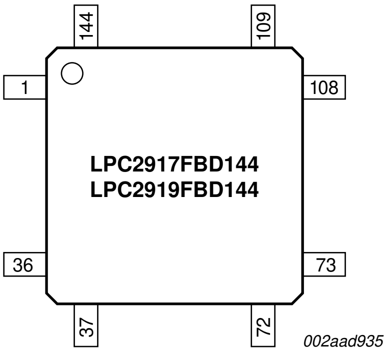


Fig 2. Pin configuration for SOT486-1 (LQFP144)

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.

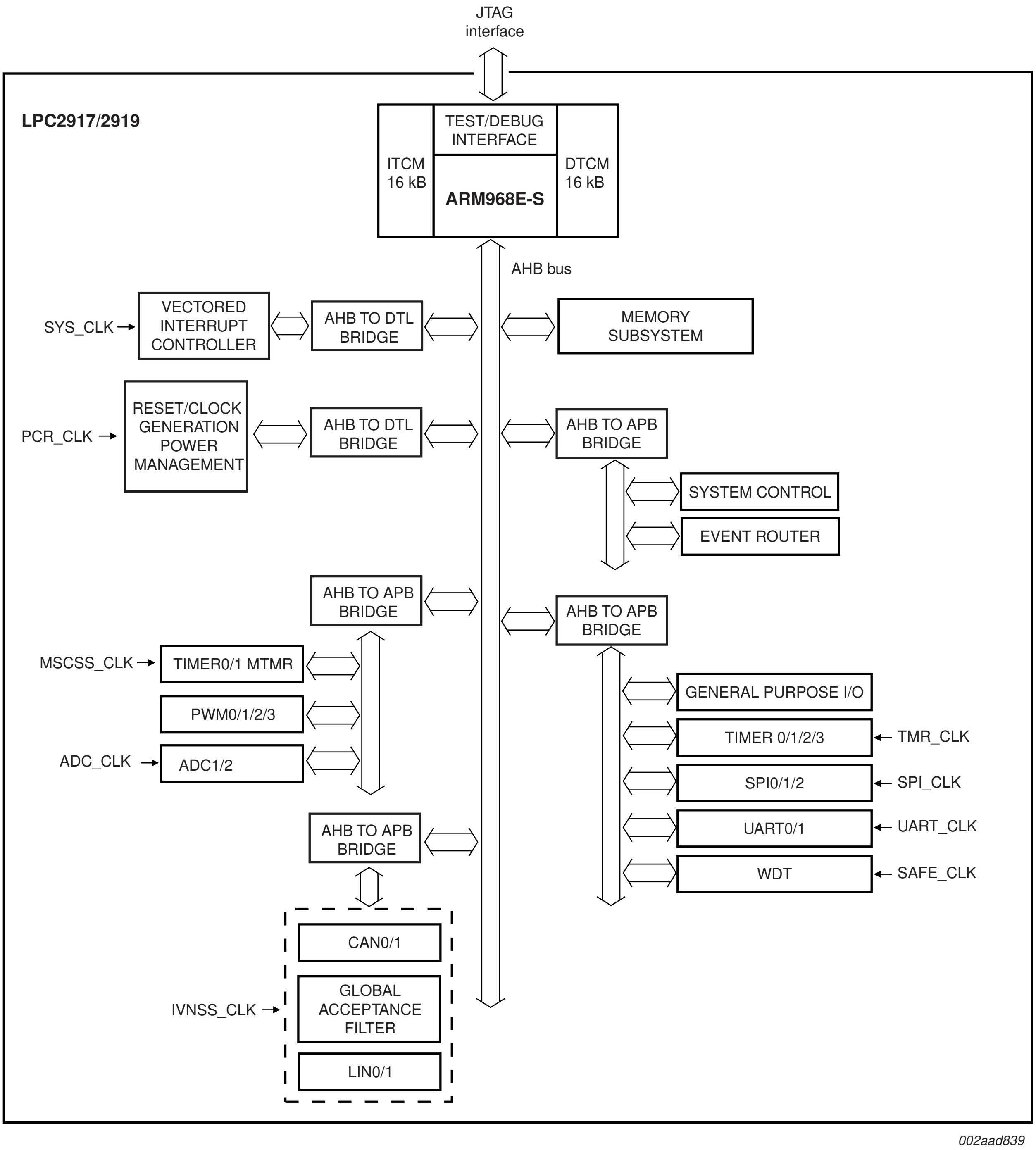


Fig 3. LPC2917/19 block diagram, overview of clock areas

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

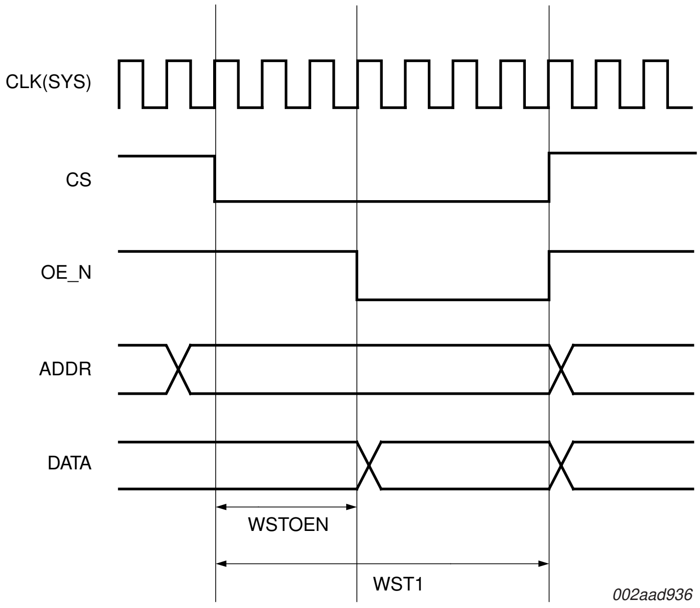
Symbol	Direction	Description
EXTBUS CSx	OUT	memory-bank x select, x runs from 0 to 7
EXTBUS BLSy	OUT	byte-lane select input y, y runs from 0 to 3
EXTBUS WE_N	OUT	write enable (active LOW)
EXTBUS OE_N	OUT	output enable (active LOW)
EXTBUS A[23:0]	OUT	address bus
EXTBUS D[31:0]	IN/OUT	data bus

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

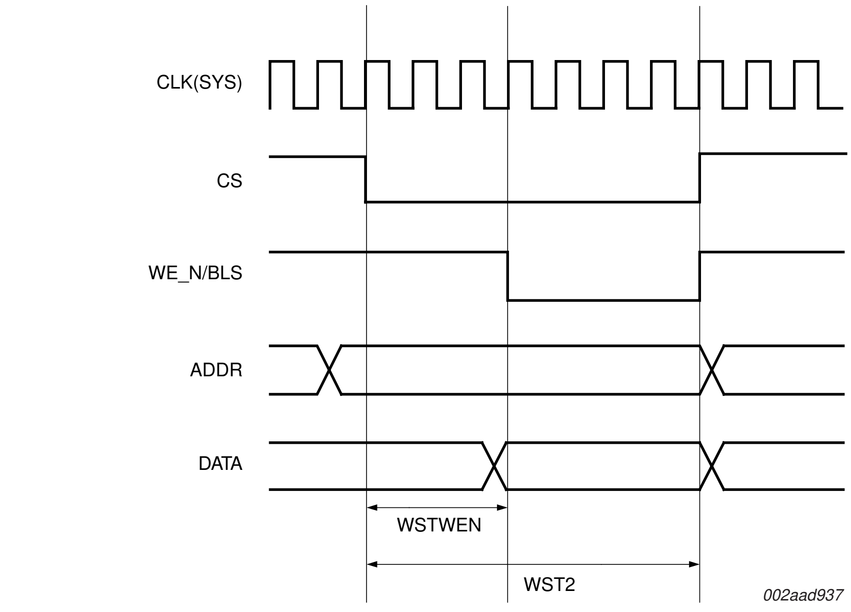
A timing diagram for reading from external memory is shown in Figure 4. The relationship between the wait-state settings is indicated with arrows.



WSTOEN = 3, WST1 = 7

Fig 4. Reading from external memory

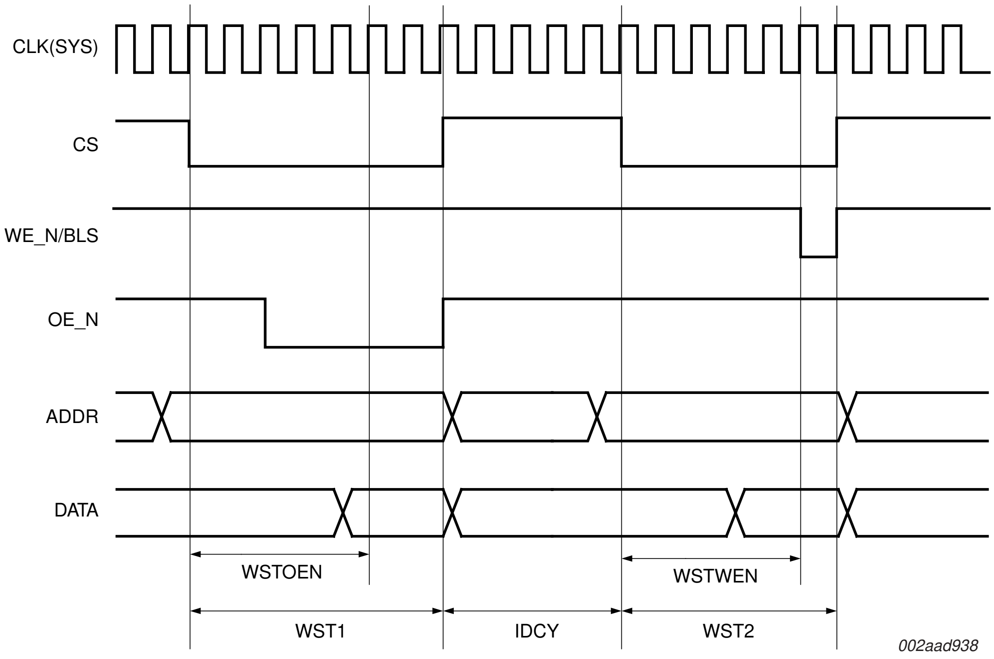
A timing diagram for writing to external memory is shown In Figure 5. The relationship between wait-state settings is indicated with arrows.



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