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

LPC2926/2927/2929

ARM9 microcontroller with CAN, LIN, and USB

Rev. 5 — 28 September 2010 Product data sheet

1. General description

The LPC2926/2927/2929 combine an ARM968E-S CPU core with two integrated TCM blocks operating at frequencies of up to 125 MHz, Full-speed USB 2.0 OTG and device controller, CAN and LIN, 56 kB SRAM, up to 768 kB flash memory, external memory interface, three 10-bit ADCs, and multiple serial and parallel interfaces in a single chip targeted at consumer, industrial and communication markets. To optimize system power consumption, the LPC2926/2927/2929 has a very flexible Clock Generation Unit (CGU) that provides dynamic clock gating and scaling.

2. Features and benefits

- ARM968E-S processor running at frequencies of up to 125 MHz maximum.
- Multi-layer AHB system bus at 125 MHz with four separate layers.
- On-chip memory:
  - Two Tightly Coupled Memories (TCM), 32 kB Instruction TCM (ITCM), 32 kB Data TCM (DTCM).
  - Two separate internal Static RAM (SRAM) instances; 32 kB SRAM and 16 kB SRAM.
  - 8 kB ETB SRAM also available for code execution and data.
  - Up to 768 kB high-speed flash-program memory.
  - 16 kB true EEPROM, byte-erasable and programmable.
- Dual-master, eight-channel GPDMA controller on the AHB multi-layer matrix which can be used with the Serial Peripheral Interface (SPI) interfaces and the UARTs, as well as for memory-to-memory transfers including the TCM memories.
- External Static Memory Controller (SMC) with eight memory banks; up to 32-bit data bus; up to 24-bit address bus.
- Serial interfaces:
  - USB 2.0 full-speed device/OTG controller with dedicated DMA controller and on-chip device PHY.
  - Two-channel CAN controller supporting FullCAN and extensive message filtering.
  - Two LIN master controllers with full hardware support for LIN communication. The LIN interface can be configured as UART to provide two additional UART interfaces.
  - Two 550 UARTs with 16-byte Tx and Rx FIFO depths, DMA support, and RS485/EIA-485 (9-bit) support.
  - Three full-duplex Q-SPIs with four slave-select lines; 16 bits wide; 8 locations deep; Tx FIFO and Rx FIFO.
  - Two I^2C-bus interfaces.

- Other peripherals:
  - One 10-bit ADC with 5.0 V measurement range and eight input channels with conversion times as low as 2.44 μs per channel.
  - Two 10-bit ADCs, 8-channels each, with 3.3 V measurement range provide an additional 16 analog inputs with conversion times as low as 2.44 μs per channel. Each channel provides a compare function to minimize interrupts.
  - Multiple trigger-start option for all ADCs: timer, PWM, other ADC, and external signal input.
  - Four 32-bit timers each containing four capture-and-compare registers linked to I/Os.
  - Four six-channel PWMs (Pulse Width Modulators) with capture and trap functionality.
  - Two dedicated 32-bit timers to schedule and synchronize PWM and ADC.
  - Quadrature encoder interface that can monitor one external quadrature encoder.
  - 32-bit watchdog with timer change protection, running on safe clock.
- Up to 104 general-purpose I/O pins with programmable pull-up, pull-down, or bus keeper.
- Vectored Interrupt Controller (VIC) with 16 priority levels.
- Up to 21 level-sensitive external interrupt pins, including USB, CAN and LIN wake-up features.
- Configurable clock-out pin for driving external system clocks.
- Processor wake-up from power-down via external interrupt pins; CAN or LIN activity.
- Flexible Reset Generator Unit (RGU) able to control resets of individual modules.
- Flexible Clock-Generation Unit (CGU0) 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. PLL input range 10 MHz to 25 MHz.
  - On-chip PLL allows CPU operation up to a maximum CPU rate of 125 MHz.
  - Generation of up to 11 base clocks.
  - Seven fractional dividers.
- Second CGU (CGU1) with its own PLL generates USB clocks and a configurable clock output.
- 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.
- ETM/ETB debug functions with 8 kB of dedicated SRAM also accessible for application code and data storage.
- 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.

3. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
LPC2926FBD144	LQFP144	plastic low profile quad flat package; 144 leads; body 20 × 20 × 1.4 mm	SOT486-1
LPC2927FBD144	LQFP144	plastic low profile quad flat package; 144 leads; body 20 × 20 × 1.4 mm	SOT486-1
LPC2929FBD144	LQFP144	plastic low profile quad flat package; 144 leads; body 20 × 20 × 1.4 mm	SOT486-1

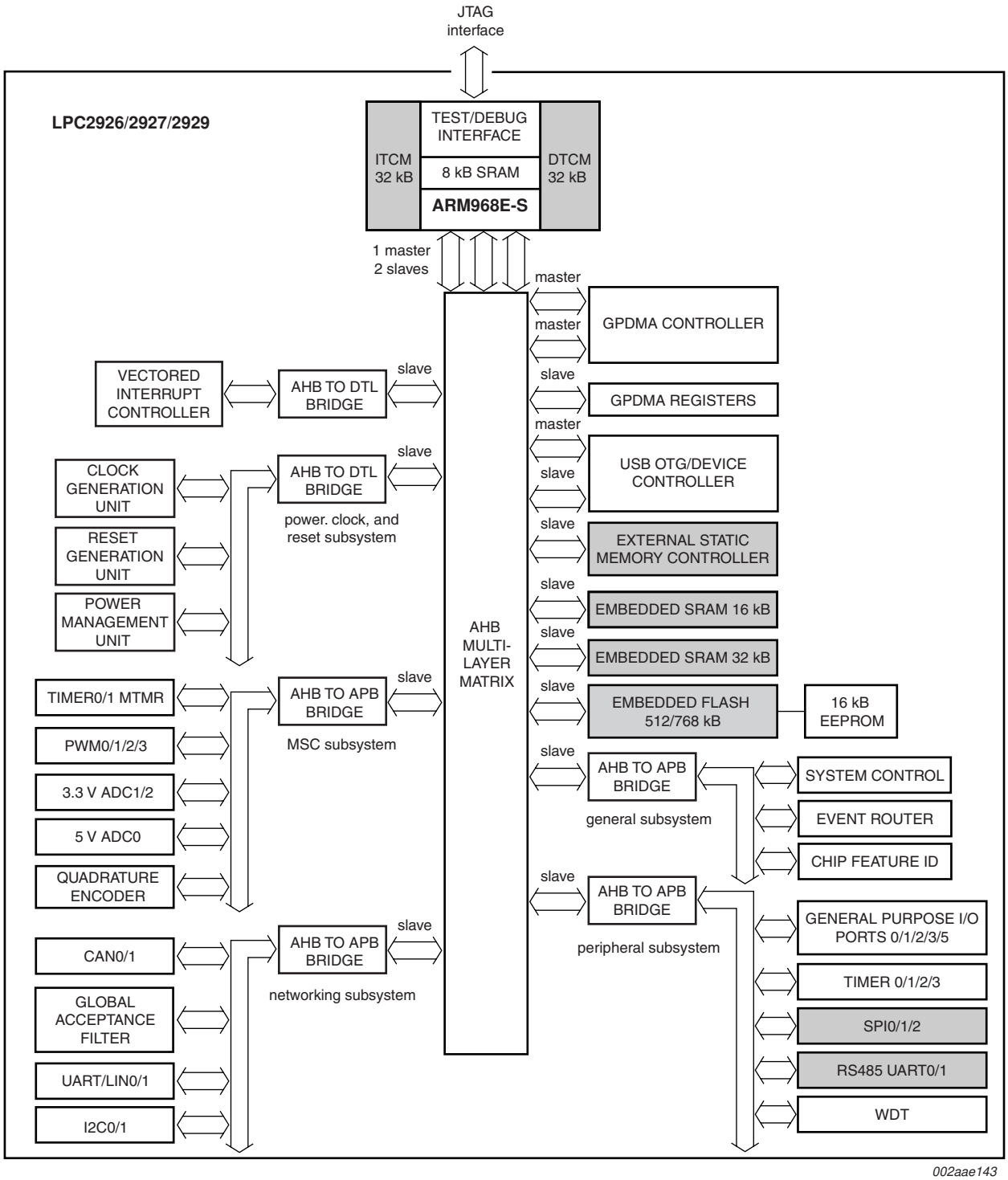
3.1 Ordering options

Table 2. Part options

Type number	Flash memory	SRAM	SMC	USB OTG/ device	UART RS485	LIN 2.0/ UART	CAN	Package
LPC2926FBD144	256 kB	56 kB + 2 × 32 kB TCM	32-bit	yes	2	2	2	LQFP144
LPC2927FBD144	512 kB	56 kB + 2 × 32 kB TCM	32-bit	yes	2	2	2	LQFP144
LPC2929FBD144	768 kB	56 kB + 2 × 32 kB TCM	32-bit	yes	2	2	2	LQFP144

[1] Note that parts LPC2926, LPC2927 and LPC2929 are not fully pin compatible with parts LPC2917, LPC2919 and LPC2917/01, LPC2919/01. The Modulation and Sampling Control SubSystem (MSCSS) and timer blocks have a reduced pinout on the LPC2926/2927/2929.

4. Block diagram



Grey-shaded blocks represent peripherals and memory regions accessible by the GPDMA.

Fig 1. LPC2926/2927/2929 block diagram

5. Pinning information

5.1 Pinning

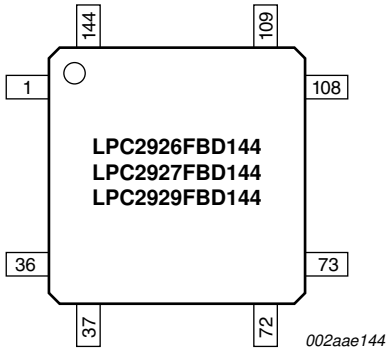


Fig 2. Pin configuration for SOT486-1 (LQFP144)

5.2 Pin description

5.2.1 General description

The LPC2926/2927/2929 uses five ports: port 0 with 32 pins, ports 1 and 2 with 28 pins each, port 3 with 16 pins, and port 5 with 2 pins. Port 4 is not used. The pin to which each function is assigned is controlled by the SFSP registers in the System Control Unit (SCU). The functions combined on each port pin are shown in the pin description tables in this section.

5.2.2 LQFP144 pin assignment

Table 3. LQFP144 pin assignment

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
TDO	1[1]	IEEE 1149.1 test data out			
P2[21]/SDI2/ PCAP2[1]/D19	2[1]	GPIO2, pin 21	SPI2 SDI	PWM2 CAP1	EXTBUS D19
P0[24]/TXD1/ TXDC1/SCS2[0]	3[1]	GPIO0, pin 24	UART1 TXD	CAN1 TXD	SPI2 SCS0
P0[25]/RXD1/ RXDC1/SDO2	4[1]	GPIO0, pin 25	UART1 RXD	CAN1 RXD	SPI2 SDO
P0[26]/TXD1/SDI2	5[1]	GPIO0, pin 26	-	UART1 TXD	SPI2 SDI
P0[27]/RXD1/SCK2	6[1]	GPIO0, pin 27	-	UART1 RXD	SPI2 SCK
P0[28]/CAP0[0]/ MAT0[0]	7[1]	GPIO0, pin 28	-	TIMER0 CAP0	TIMER0 MAT0
P0[29]/CAP0[1]/ MAT0[1]	8[1]	GPIO0, pin 29	-	TIMER0 CAP1	TIMER0 MAT1
$V_{DD(IO)}$	9	3.3 V power supply for I/O			

Table 3. LQFP144 pin assignment *...continued*

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
P2[22]/SCK2/ PCAP2[2]/D20	10[1]	GPIO2, pin 22	SPI2 SCK	PWM2 CAP2	EXTBUS D20
P2[23]/SCS1[0]/ PCAP3[0]/D21	11[1]	GPIO2, pin 23	SPI1 SCS0	PWM3 CAP0	EXTBUS D21
P3[6]/SCS0[3]/ PMAT1[0]/TXDL1	12[1]	GPIO3, pin 6	SPI0 SCS3	PWM1 MAT0	LIN1/UART TXD
P3[7]/SCS2[1]/ PMAT1[1]/RXDL1	13[1]	GPIO3, pin 7	SPI2 SCS1	PWM1 MAT1	LIN1/UART RXD
P0[30]/CAP0[2]/ MAT0[2]	14[1]	GPIO0, pin 30	-	TIMER0 CAP2	TIMER0 MAT2
P0[31]/CAP0[3]/ MAT0[3]	15[1]	GPIO0, pin 31	-	TIMER0 CAP3	TIMER0 MAT3
P2[24]/SCS1[1]/ PCAP3[1]/D22	16[1]	GPIO2, pin 24	SPI1 SCS1	PWM3 CAP1	EXTBUS D22
P2[25]/SCS1[2]/ PCAP3[2]/D23	17[1]	GPIO2, pin 25	SPI1 SCS2	PWM3 CAP2	EXTBUS D23
$V_{SS(IO)}$	18	ground for I/O			
P5[19]/USB_D+	19[2]	GPIO5, pin 19	USB_D+	-	-
P5[18]/USB_D–	20[2]	GPIO5, pin 18	USB_D–	-	-
$V_{DD(IO)}$	21	3.3 V power supply for I/O			
$V_{DD(CORE)}$	22	1.8 V power supply for digital core			
$V_{SS(CORE)}$	23	ground for core			
$V_{SS(IO)}$	24	ground for I/O			
P3[8]/SCS2[0]/ PMAT1[2]	25[1]	GPIO3, pin 8	SPI2 SCS0	PWM1 MAT2	-
P3[9]/SDO2/ PMAT1[3]	26[1]	GPIO3, pin 9	SPI2 SDO	PWM1 MAT3	-
P2[26]/CAP0[2]/ MAT0[2]/EI6	27[1]	GPIO2, pin 26	TIMER0 CAP2	TIMER0 MAT2	EXTINT6
P2[27]/CAP0[3]/ MAT0[3]/EI7	28[1]	GPIO2, pin 27	TIMER0 CAP3	TIMER0 MAT3	EXTINT7
P1[27]/CAP1[2]/ TRAP2/PMAT3[3]	29[1]	GPIO1, pin 27	TIMER1 CAP2, ADC2 EXT START	PWM TRAP2	PWM3 MAT3
P1[26]/PMAT2[0]/ TRAP3/PMAT3[2]	30[1]	GPIO1, pin 26	PWM2 MAT0	PWM TRAP3	PWM3 MAT2
$V_{DD(IO)}$	31	3.3 V power supply for I/O			
P1[25]/PMAT1[0]/ USB_VBUS/ PMAT3[1]	32[1]	GPIO1, pin 25	PWM1 MAT0	USB_VBUS	PWM3 MAT1
P1[24]/PMAT0[0]/ $\overline{\text{USB\_CONNECT}}$/ PMAT3[0]	33[1]	GPIO1, pin 24	PWM0 MAT0	$\overline{\text{USB\_CONNECT}}$	PWM3 MAT0
P1[23]/RXD0/ USB_SSPND/$\overline{\text{CS5}}$	34[1]	GPIO1, pin 23	UART0 RXD	USB_SSPND	EXTBUS $\overline{\text{CS5}}$

Table 3. LQFP144 pin assignment *...continued*

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
P1[22]/TXD0/ USB_UP_LED/ CS4	35[1]	GPIO1, pin 22	UART0 TXD	USB_UP_LED	EXTBUS CS4
TMS	36[1]	IEEE 1149.1 test mode select, pulled up internally			
TCK	37[1]	IEEE 1149.1 test clock			
P1[21]/CAP3[3]/ CAP1[3]/D7	38[1]	GPIO1, pin 21	TIMER3 CAP3	TIMER1 CAP3, MSCSS PAUSE	EXTBUS D7
P1[20]/CAP3[2]/ SCS0[1]/D6	39[1]	GPIO1, pin 20	TIMER3 CAP2	SPI0 SCS1	EXTBUS D6
P1[19]/CAP3[1]/ SCS0[2]/D5	40[1]	GPIO1, pin 19	TIMER3 CAP1	SPI0 SCS2	EXTBUS D5
P1[18]/CAP3[0]/ SDO0/D4	41[1]	GPIO1, pin 18	TIMER3 CAP0	SPI0 SDO	EXTBUS D4
P1[17]/CAP2[3]/ SDI0/D3	42[1]	GPIO1, pin 17	TIMER2 CAP3	SPI0 SDI	EXTBUS D3
$V_{SS(IO)}$	43	ground for I/O			
P1[16]/CAP2[2]/ SCK0/D2	44[1]	GPIO1, pin 16	TIMER2 CAP2	SPI0 SCK	EXTBUS D2
P2[0]/MAT2[0]/ TRAP3/D8	45[1]	GPIO2, pin 0	TIMER2 MAT0	PWM TRAP3	EXTBUS D8
P2[1]/MAT2[1]/ TRAP2/D9	46[1]	GPIO2, pin 1	TIMER2 MAT1	PWM TRAP2	EXTBUS D9
P3[10]/SDI2/ PMAT1[4]	47[1]	GPIO3, pin 10	SPI2 SDI	PWM1 MAT4	-
P3[11]/SCK2/ PMAT1[5]/USB_LS	48[1]	GPIO3, pin 11	SPI2 SCK	PWM1 MAT5	USB_LS
P1[15]/CAP2[1]/ SCS0[0]/D1	49[1]	GPIO1, pin 15	TIMER2 CAP1	SPI0 SCS0	EXTBUS D1
P1[14]/CAP2[0]/ SCS0[3]/D0	50[1]	GPIO1, pin 14	TIMER2 CAP0	SPI0 SCS3	EXTBUS D0
P1[13]/SCL1/ EI3/WE	51[1]	GPIO1, pin 13	EXTINT3	I2C1 SCL	EXTBUS WE
P1[12]/SDA1/ EI2/OE	52[1]	GPIO1, pin 12	EXTINT2	I2C1 SDA	EXTBUS OE
$V_{DD(IO)}$	53	3.3 V power supply for I/O			
P2[2]/MAT2[2]/ TRAP1/D10	54[1]	GPIO2, pin 2	TIMER2 MAT2	PWM TRAP1	EXTBUS D10
P2[3]/MAT2[3]/ TRAP0/D11	55[1]	GPIO2, pin 3	TIMER2 MAT3	PWM TRAP0	EXTBUS D11
P1[11]/SCK1/ SCL0/CS3	56[1]	GPIO1, pin 11	SPI1 SCK	I2C0 SCL	EXTBUS CS3
P1[10]/SDI1/ SDA0/CS2	57[1]	GPIO1, pin 10	SPI1 SDI	I2C0 SDA	EXTBUS CS2
P3[12]/SCS1[0]/EI4/ USB_SSPND	58[1]	GPIO3, pin 12	SPI1 SCS0	EXTINT4	USB_SSPND

Table 3. LQFP144 pin assignment *...continued*

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
$V_{SS(CORE)}$	59	ground for digital core			
$V_{DD(CORE)}$	60	1.8 V power supply for digital core			
P3[13]/SDO1/ EI5/IDX0	61[1]	GPIO3, pin 13	SPI1 SDO	EXTINT5	QEI0 IDX
P2[4]/MAT1[0]/ EI0/D12	62[1]	GPIO2, pin 4	TIMER1 MAT0	EXTINT0	EXTBUS D12
P2[5]/MAT1[1]/ EI1/D13	63[1]	GPIO2, pin 5	TIMER1 MAT1	EXTINT1	EXTBUS D13
P1[9]/SDO1/ RXDL1/$\overline{CS1}$	64[1]	GPIO1, pin 9	SPI1 SDO	LIN1 RXD/UART RXD	EXTBUS $\overline{CS1}$
$V_{SS(IO)}$	65	ground for I/O			
P1[8]/SCS1[0]/ TXDL1/$\overline{CS0}$	66[1]	GPIO1, pin 8	SPI1 SCS0	LIN1 TXD/UART TXD	EXTBUS $\overline{CS0}$
P1[7]/SCS1[3]/RXD1/ A7	67[1]	GPIO1, pin 7	SPI1 SCS3	UART1 RXD	EXTBUS A7
P1[6]/SCS1[2]/ TXD1/A6	68[1]	GPIO1, pin 6	SPI1 SCS2	UART1 TXD	EXTBUS A6
P2[6]/MAT1[2]/ EI2/D14	69[1]	GPIO2, pin 6	TIMER1 MAT2	EXTINT2	EXTBUS D14
P1[5]/SCS1[1]/PMAT 3[5]/A5	70[1]	GPIO1, pin 5	SPI1 SCS1	PWM3 MAT5	EXTBUS A5
P1[4]/SCS2[2]/PMAT 3[4]/A4	71[1]	GPIO1, pin 4	SPI2 SCS2	PWM3 MAT4	EXTBUS A4
$\overline{TRST}$	72[1]	IEEE 1149.1 test reset NOT; active LOW; pulled up internally			
$\overline{RST}$	73[1]	asynchronous device reset; active LOW; pulled up internally			
$V_{SS(OSC)}$	74	ground for oscillator			
XOUT_OSC	75[3]	crystal out for oscillator			
XIN_OSC	76[3]	crystal in for oscillator			
$V_{DD(OSC\_PLL)}$	77	1.8 V supply for oscillator and PLL			
$V_{SS(PLL)}$	78	ground for PLL			
P2[7]/MAT1[3]/ EI3/D15	79[1]	GPIO2, pin 7	TIMER1 MAT3	EXTINT3	EXTBUS D15
P3[14]/SDI1/ EI6/TXDC0	80[1]	GPIO3, pin 14	SPI1 SDI	EXTINT6	CAN0 TXD
P3[15]/SCK1/ EI7/RXDC0	81[1]	GPIO3, pin 15	SPI1 SCK	EXTINT7	CAN0 RXD
$V_{DD(IO)}$	82	3.3 V power supply for I/O			
P2[8]/CLK_OUT/ PMAT0[0]/SCS0[2]	83[1]	GPIO2, pin 8	CLK_OUT	PWM0 MAT0	SPI0 SCS2
P2[9]/ $\overline{USB\_UP\_LED}$/ PMAT0[1]/ SCS0[1]	84[1]	GPIO2, pin 9	$\overline{USB\_UP\_LED}$	PWM0 MAT1	SPI0 SCS1

Table 3. LQFP144 pin assignment *...continued*

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
P1[3]/SCS2[1]/ PMAT3[3]/A3	85[1]	GPIO1, pin 3	SPI2 SCS1	PWM3 MAT3	EXTBUS A3
P1[2]/SCS2[3]/ PMAT3[2]/A2	86[1]	GPIO1, pin 2	SPI2 SCS3	PWM3 MAT2	EXTBUS A2
P1[1]/EI1/ PMAT3[1]/A1	87[1]	GPIO1, 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[1]	GPIO1, pin 0	EXTINT0	PWM3 MAT0	EXTBUS A0
P2[10]/ PMAT0[2]/ SCS0[0]	91[1]	GPIO2, pin 10	$\overline{\text{USB\_INT}}$	PWM0 MAT2	SPI0 SCS0
P2[11]/ PMAT0[3]/SCK0	92[1]	GPIO2, pin 11	$\overline{\text{USB\_RST}}$	PWM0 MAT3	SPI0 SCK
P0[0]/PHB0/ TXDC0/D24	93[1]	GPIO0, pin 0	QEI0 PHB	CAN0 TXD	EXTBUS D24
$V_{SS(IO)}$	94	ground for I/O			
P0[1]/PHA0/ RXDC0/D25	95[1]	GPIO0, pin 1	QEI 0 PHA	CAN0 RXD	EXTBUS D25
P0[2]/CLK_OUT/ PMAT0[0]/D26	96[1]	GPIO0, pin 2	CLK_OUT	PWM0 MAT0	EXTBUS D26
P0[3]/$\overline{\text{USB\_UP\_LED}}$/ PMAT0[1]/D27	97[1]	GPIO0, pin 3	$\overline{\text{USB\_UP\_LED}}$	PWM0 MAT1	EXTBUS D27
P3[0]/IN0[6]/ PMAT2[0]/$\overline{\text{CS6}}$	98[1]	GPIO3, pin 0	ADC0 IN6	PWM2 MAT0	EXTBUS $\overline{\text{CS6}}$
P3[1]/IN0[7/ PMAT2[1]/$\overline{\text{CS7}}$	99[1]	GPIO3, pin 1	ADC0 IN7	PWM2 MAT1	EXTBUS $\overline{\text{CS7}}$
P2[12]/IN0[4] PMAT0[4]/SDI0	100[1]	GPIO2, pin 12	ADC0 IN4	PWM0 MAT4	SPI0 SDI
P2[13]/IN0[5] PMAT0[5]/SDO0	101[1]	GPIO2, pin 13	ADC0 IN5	PWM0 MAT5	SPI0 SDO
P0[4]/IN0[0]/ PMAT0[2]/D28	102[1]	GPIO0, pin 4	ADC0 IN0	PWM0 MAT2	EXTBUS D28
P0[5]/IN0[1]/ PMAT0[3]/D29	103[1]	GPIO0, pin 5	ADC0 IN1	PWM0 MAT3	EXTBUS D29
$V_{DD(IO)}$	104	3.3 V power supply for I/O			
P0[6]/IN0[2]/ PMAT0[4]/D30	105[1]	GPIO0, pin 6	ADC0 IN2	PWM0 MAT4	EXTBUS D30
P0[7]/IN0[3]/ PMAT0[5]/D31	106[1]	GPIO0, pin 7	ADC0 IN3	PWM0 MAT5	EXTBUS D31
$V_{DDA(ADC3V3)}$	107	3.3 V power supply for ADC			
JTAGSEL	108[1]	TAP controller select input; LOW-level selects the ARM debug mode; HIGH-level selects boundary scan; pulled up internally.			

Table 3. LQFP144 pin assignment *...continued*

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
$V_{DDA(ADC5V0)}$	109	5 V supply voltage for ADC0 and 5 V reference for ADC0.			
VREFP	110[3]	HIGH reference for ADC			
VREFN	111[3]	LOW reference for ADC			
P0[8]/IN1[0]/TXDL0/ A20	112[4]	GPIO0, pin 8	ADC1 IN0	LIN0 TXD/UART TXD	EXTBUS A20
P0[9]/IN1[1]/ RXDL0/A21	113[4]	GPIO0, pin 9	ADC1 IN1	LIN0 RXD/UART TXD	EXTBUS A21
P0[10]/IN1[2]/ PMAT1[0]/A8	114[4]	GPIO0, pin 10	ADC1 IN2	PWM1 MAT0	EXTBUS A8
P0[11]/IN1[3]/ PMAT1[1]/A9	115[4]	GPIO0, pin 11	ADC1 IN3	PWM1 MAT1	EXTBUS A9
P2[14]/SDA1/ PCAP0[0]/$\overline{BLS0}$	116[1]	GPIO2, pin 14	I2C1 SDA	PWM0 CAP0	EXTBUS $\overline{BLS0}$
P2[15]/SCL1/ PCAP0[1]/$\overline{BLS1}$	117[1]	GPIO2, pin 15	I2C1 SCL	PWM0 CAP1	EXTBUS $\overline{BLS1}$
P3[2]/MAT3[0]/ PMAT2[2]/ USB_SDA	118[1]	GPIO3, pin 2	TIMER3 MAT0	PWM2 MAT2	USB_SDA
$V_{SS(IO)}$	119	ground for I/O			
P3[3]/MAT3[1]/ PMAT2[3]/ USB_SCL	120[1]	GPIO3, pin 3	TIMER3 MAT1	PWM2 MAT3	USB_SCL
P0[12]/IN1[4]/ PMAT1[2]/A10	121[4]	GPIO0, pin 12	ADC1 IN4	PWM1 MAT2	EXTBUS A10
P0[13]/IN1[5]/ PMAT1[3]/A11	122[4]	GPIO0, pin 13	ADC1 IN5	PWM1 MAT3	EXTBUS A11
P0[14]/IN1[6]/ PMAT1[4]/A12	123[4]	GPIO0, pin 14	ADC1 IN6	PWM1 MAT4	EXTBUS A12
P0[15]/IN1[7]/ PMAT1[5]/A13	124[4]	GPIO0, pin 15	ADC1 IN7	PWM1 MAT5	EXTBUS A13
P0[16]IN2[0]/ TXD0/A22	125[4]	GPIO0, pin 16	ADC2 IN0	UART0 TXD	EXTBUS A22
P0[17]/IN2[1]/ RXD0/A23	126[4]	GPIO0, 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]/$\overline{BLS2}$	129[1]	GPIO2, pin 16	UART1 TXD	PWM0 CAP2	EXTBUS $\overline{BLS2}$
P2[17]/RXD1/ PCAP1[0]/$\overline{BLS3}$	130[1]	GPIO2, pin 17	UART1 RXD	PWM1 CAP0	EXTBUS $\overline{BLS3}$
$V_{DD(IO)}$	131	3.3 V power supply for I/O			
P0[18]/IN2[2]/ PMAT2[0]/A14	132[4]	GPIO0, pin 18	ADC2 IN2	PWM2 MAT0	EXTBUS A14

Table 3. LQFP144 pin assignment *...continued*

Pin name	Pin	Description			
		Function 0 (default)	Function 1	Function 2	Function 3
P0[19]/IN2[3]/ PMAT2[1]/A15	133[4]	GPIO0, pin 19	ADC2 IN3	PWM2 MAT1	EXTBUS A15
P3[4]/MAT3[2]/ PMAT2[4]/TXDC1	134[1]	GPIO3, pin 4	TIMER3 MAT2	PWM2 MAT4	CAN1 TXD
P3[5]/MAT3[3]/ PMAT2[5]/RXDC1	135[1]	GPIO3, pin 5	TIMER3 MAT3	PWM2 MAT5	CAN1 RXD
P2[18]/SCS2[1]/ PCAP1[1]/D16	136[1]	GPIO2, pin 18	SPI2 SCS1	PWM1 CAP1	EXTBUS D16
P2[19]/SCS2[0]/ PCAP1[2]/D17	137[1]	GPIO2, pin 19	SPI2 SCS0	PWM1 CAP2	EXTBUS D17
P0[20]/IN2[4]/ PMAT2[2]/A16	138[4]	GPIO0, pin 20	ADC2 IN4	PWM2 MAT2	EXTBUS A16
P0[21]/IN2[5]/ PMAT2[3]/A17	139[4]	GPIO0, pin 21	ADC2 IN5	PWM2 MAT3	EXTBUS A17
P0[22]/IN2[6]/ PMAT2[4]/A18	140[4]	GPIO0, pin 22	ADC2 IN6	PWM2 MAT4	EXTBUS A18
$V_{SS(IO)}$	141	ground for I/O			
P0[23]/IN2[7]/ PMAT2[5]/A19	142[4]	GPIO0, pin 23	ADC2 IN7	PWM2 MAT5	EXTBUS A19
P2[20]/ PCAP2[0]/D18	143[1]	GPIO2, pin 20	SPI2 SDO	PWM2 CAP0	EXTBUS D18
TDI	144[1]	IEEE 1149.1 data in, pulled up internally			

[1] Bidirectional pad; analog port; plain input; 3-state output; slew rate control; 5 V tolerant; TTL with hysteresis; programmable pull-up/pull-down/repeater.

[2] USB pad.

[3] Analog pad; analog I/O.

[4] Analog I/O pad.

6. Functional description

6.1 Architectural overview

The LPC2926/2927/2929 consists of:

- An ARM968E-S processor with real-time emulation support
- An AMBA multi-layer Advanced High-performance Bus (AHB) for interfacing to the on-chip memory controllers
- Two DTL buses (an 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 LPC2926/2927/2929 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 AHB-to-APB 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.

6.2 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 Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed Complex Instruction Set Computers (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.

6.3 On-chip flash memory system

The LPC2926/2927/2929 includes a 256 kB, 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 via the flash memory controller or the JTAG.

The flash controller also supports a 16 kB, byte-accessible on-chip EEPROM integrated on the LPC2926/2927/2929.

6.4 On-chip static RAM

In addition to the two 32 kB TCMs the LPC2926/2927/2929 includes two static RAM memories: one of 32 kB and one of 16 kB. Both may be used for code and/or data storage.

In addition, 8 kB SRAM for the ETB can be used as static memory for code and data storage. However, DMA access to this memory region is not supported.

6.5 Memory map

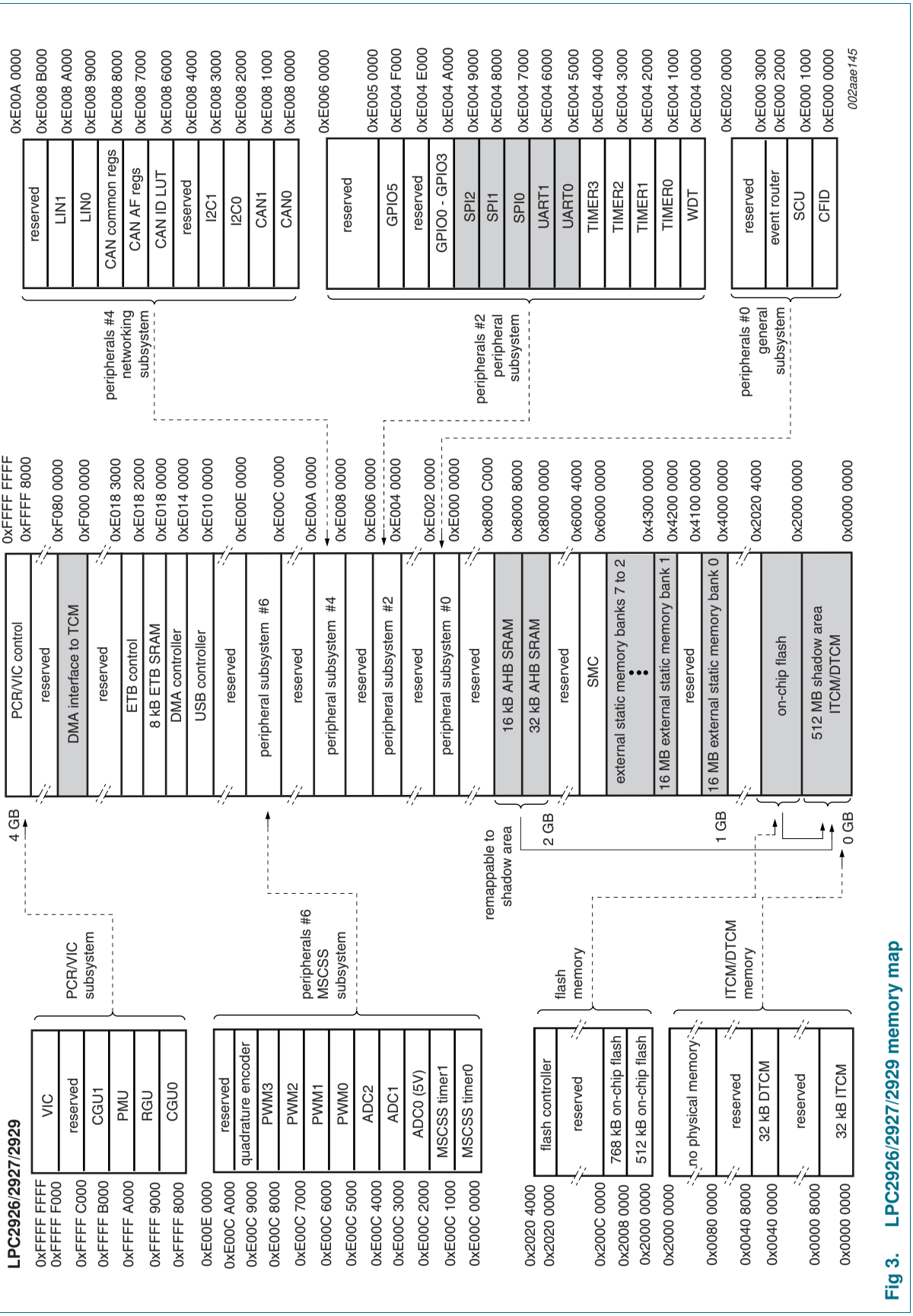


Fig 3. LPC2926/2927/2929 memory map

6.6 Reset, debug, test, and power description

6.6.1 Reset and power-up behavior

The LPC2926/2927/2929 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 8 for trip levels of the internal power-up reset circuit[1]. See Section 9 for characteristics of the several start-up and initialization times. Table 4 shows the reset pin.

Table 4. Reset pin

Symbol	Direction	Description
$\overline{RST}$	IN	external reset input, active LOW; pulled up internally

At activation of the $\overline{RST}$ pin the JTAGSEL pin is sensed as logic LOW. If this is the case the LPC2926/2927/2929 is assumed to be connected to debug hardware, and internal circuits re-program 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.

6.6.2 Reset strategy

The LPC2926/2927/2929 contains a central module, the Reset Generator 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.

6.6.3 IEEE 1149.1 interface pins (JTAG boundary scan test)

The LPC2926/2927/2929 contains boundary-scan test logic according to IEEE 1149.1, also referred to in this document as Joint Test Action Group (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
$\overline{TRST}$	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

1. Only for 1.8 V power sources

6.6.3.1 ETM/ETB

The ETM provides real-time trace capability for deeply embedded processor cores. It outputs information about processor execution to a trace buffer. A software debugger allows configuration of the ETM using a JTAG interface and displays the trace information that has been captured in a format that a user can easily understand. The ETB stores trace data produced by the ETM.

The ETM/ETB module has the following features:

- Closely tracks the instructions that the ARM core is executing.
- On-chip trace data storage (ETB).
- All registers are programmed through JTAG interface.
- Does not consume power when trace is not being used.
- THUMB/Java instruction set support.

6.6.4 Power supply pins

Table 6 shows the power supply pins.

Table 6. Power supply pins

Symbol	Description
$V_{DD(CORE)}$	digital core supply 1.8 V
$V_{SS(CORE)}$	digital core ground (digital core, ADC0/1/2)
$V_{DD(IO)}$	I/O pins supply 3.3 V
$V_{SS(IO)}$	I/O pins ground
$V_{DD(OSC\_PLL)}$	oscillator and PLL supply
$V_{SS(OSC)}$	oscillator ground
$V_{SS(PLL)}$	PLL ground
$V_{DDA(ADC3V3)}$	ADC1 and ADC2 3.3 V supply
$V_{DDA(ADC5V0)}$	ADC0 5.0 V supply

6.7 Clocking strategy

6.7.1 Clock architecture

The LPC2926/2927/2929 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 Generator Unit (CGU0). 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 4 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 6.16 for more details of clock and power control within the device.

Two of the base clocks generated by the CGU0 are used as input into a second, dedicated CGU (CGU1). The CGU1 uses its own PLL and fractional dividers to generate two base clocks for the USB controller and one base clock for an independent clock output.

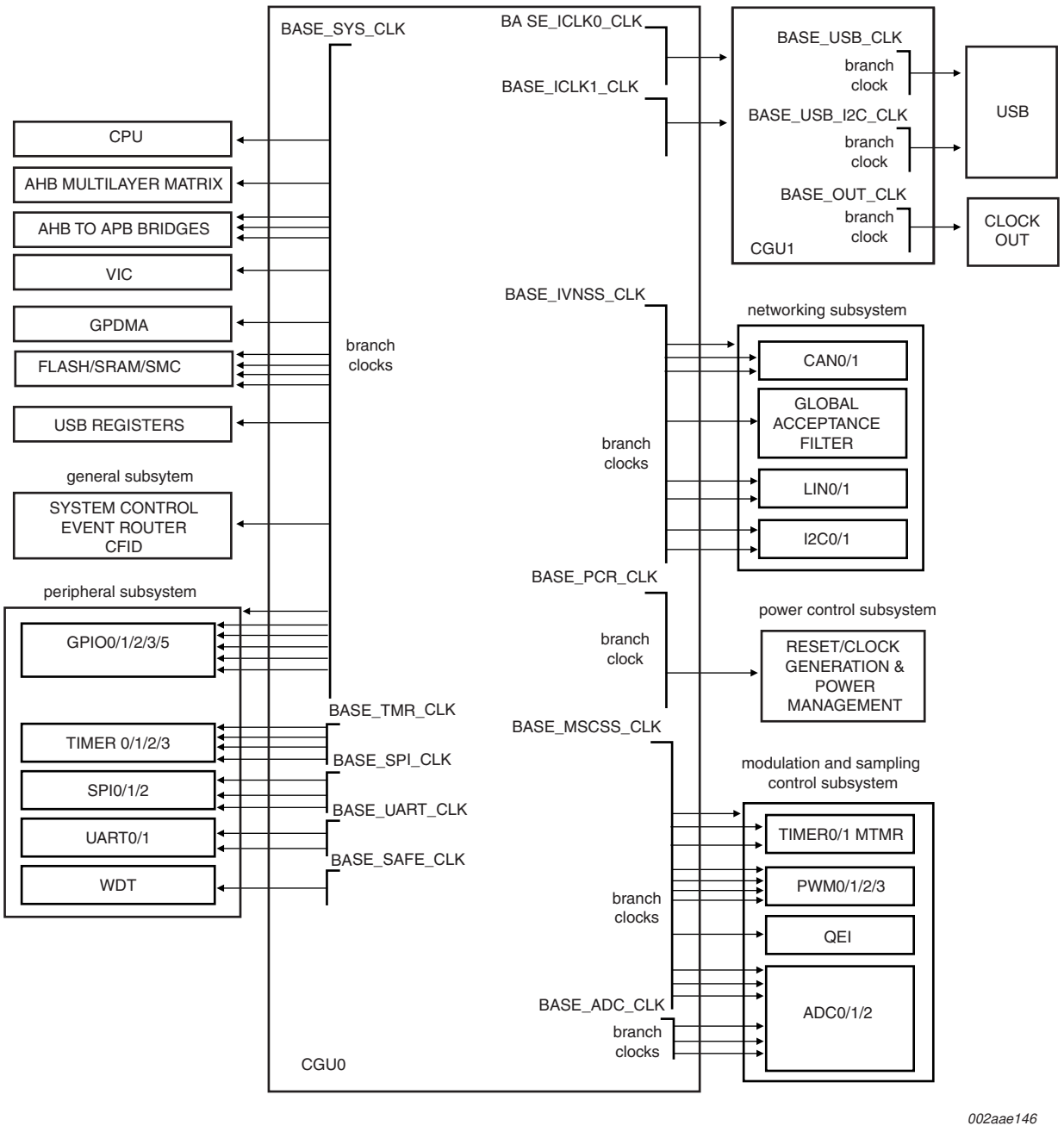


Fig 4. LPC2926/2927/2929 overview of clock areas

6.7.2 Base clock and branch clock relationship

Table 7 contains an overview of all the base blocks in the LPC2926/2927/2929 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 not be switched off. See Section 6.16.5 for more details of how to control the individual branch clocks.

Table 7. CGU0 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][3]
	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_GPIO5	GPIO bank 5	
	CLK_SYS_IVNSS_A	AHB side of bridge of IVNSS	
	CLK_SYS_MSCSS_A	AHB side of bridge of MSCSS	
	CLK_SYS_DMA	GPDMA	
	CLK_SYS_USB	USB registers	
BASE_PCR_CLK	CLK_PCR_SLOW	PCRSS, CGU, RGU and PMU logic clock	[1][4]
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_I2C0	I2C0	
	CLK_IVNSS_I2C1	I2C1	
	CLK_IVNSS_LIN0	LIN channel 0	
	CLK_IVNSS_LIN1	LIN channel 1	

Table 7. CGU0 base clock and branch clock overview *...continued*

Base clock	Branch clock name	Parts of the device clocked by this branch clock	Remark
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 1	
	CLK_MSCSS_PWM2	PWM 2	
	CLK_MSCSS_PWM3	PWM 3	
	CLK_MSCSS_ADC0_APB	APB side of ADC 0	
	CLK_MSCSS_ADC1_APB	APB side of ADC 1	
	CLK_MSCSS_ADC2_APB	APB side of ADC 2	
	CLK_MSCSS_QEI	Quadrature encoder	
BASE_UART_CLK	CLK_UART0	UART 0 interface clock	
	CLK_UART1	UART 1 interface clock	
BASE_ICLK0_CLK	-	CGU1 input 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_ADC0	Control of ADC 0, capture sample result	
	CLK_ADC1	Control of ADC 1, capture sample result	
	CLK_ADC2	Control of ADC 2, capture sample result	
reserved	-	-	
BASE_ICLK1_CLK	-	CGU1 input clock	

[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 6.13 for details.

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

[4] In the Power Clock and Reset Control subsystem parts of the CGU, RGU, and PMU have their own clock source. See Section 6.16 for details.

Table 8. CGU1 base clock and branch clock overview

Base clock	Branch clock name	Parts of the device clocked by this branch clock	Remark
BASE_OUT_CLK	CLK_OUT_CLK	clock out pin	
BASE_USB_CLK	CLK_USB_CLK	USB clock	
BASE_USB_I2C_CLK	CLK_USB_I2C_CLK	USB OTG I2C clock	

6.8 Flash memory controller

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. Flash memory contents can be protected by disabling JTAG access. Suspension of burning or erasing is not supported.

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

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

The key features are:

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

6.8.1 Functional description

After reset, flash initialization is started, which takes t_{init} time (see Section 9). 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. In this case, ARM debug facilities are disabled and flash memory contents cannot be read. If sector security is active, only the unsecured sections can be 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 9.

Table 9. 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

6.8.2 Pin description

The flash memory controller has no external pins. However, the flash can be programmed via the JTAG pins, see Section 6.6.3.

6.8.3 Clock description

The flash memory controller is clocked by CLK_SYS_FMC, see Section 6.7.2.

6.8.4 Flash layout

The ARM processor can program the flash for ISP (In-System Programming) through the flash memory controller. 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 10 gives an overview of the flash-sector base addresses.

Table 10. Flash sector overview

Sector number	Sector size (kB)	Sector base address
11	8	0x2000 0000
12	8	0x2000 2000
13	8	0x2000 4000
14	8	0x2000 6000
15	8	0x2000 8000
16	8	0x2000 A000
17	8	0x2000 C000
18	8	0x2000 E000
0	64	0x2001 0000
1	64	0x2002 0000
2	64	0x2003 0000
3[1]	64	0x2004 0000
4[1]	64	0x2005 0000
5[1]	64	0x2006 0000
6[1]	64	0x2007 0000
7[1]	64	0x2008 0000
8[1]	64	0x2009 0000
9[1]	64	0x200A 0000
10[1]	64	0x200B 0000

[1] Availability of sector 3 to sector 10 depends on device type, see Section 3 “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, once programmed, cannot be erased. Any flash operation must be executed out of SRAM (internal or external).

6.8.5 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:

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

Asynchronous reading:

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

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.

6.8.6 EEPROM

EEPROM is a non-volatile memory mostly used for storing relatively small amounts of data, for example for storing settings. It contains one 16 kB memory block and is byte-programmable and byte-erasable.

The EEPROM can be accessed only through the flash controller.

6.9 External static memory controller

The LPC2926/2927/2929 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

6.9.1 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 11 shows how the 32-bit system address is mapped to the external bus memory base addresses, chip selects, and bank internal addresses.

Table 11. 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 Ref. 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 12. 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

6.9.2 Pin description

The external static-memory controller module in the LPC2926/2927/2929 has the following pins, which are combined with other functions on the port pins of the LPC2926/2927/2929. Table 13 shows the external memory controller pins.

Table 13. External memory controller pins

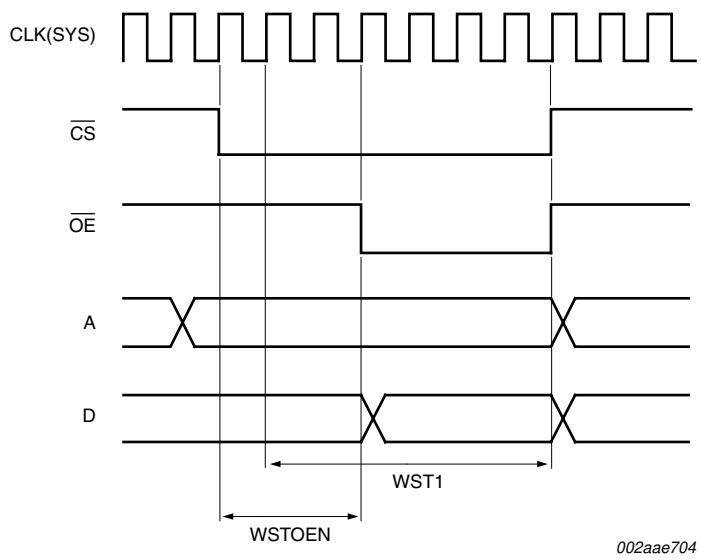
Symbol	Pin name	Direction	Description
EXTBUS $\overline{CSx}$	CSx	OUT	memory-bank x select, x runs from 0 to 7
EXTBUS $\overline{BLSy}$	$\overline{BLSy}$	OUT	byte-lane select input y, y runs from 0 to 3
EXTBUS $\overline{WE}$	$\overline{WE}$	OUT	write enable (active LOW)
EXTBUS $\overline{OE}$	OE	OUT	output enable (active LOW)
EXTBUS A[23:0]	A[23:0]	OUT	address bus
EXTBUS D[31:0]	D[31:0]	IN/OUT	data bus

6.9.3 Clock description

The External Static Memory Controller is clocked by CLK_SYS_SMC, see Section 6.7.2.

6.9.4 External memory timing diagrams

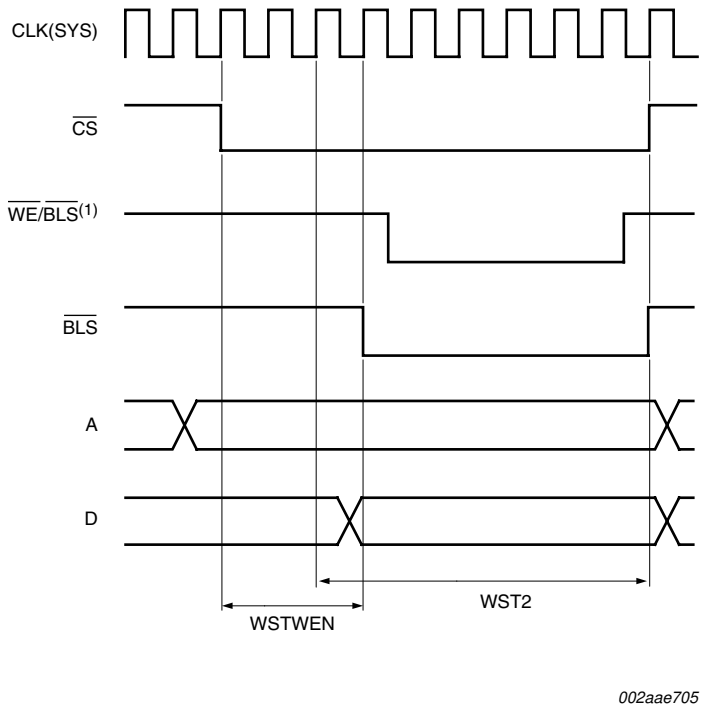
A timing diagram for reading from external memory is shown in Figure 5. The relationship between the wait-state settings is indicated with arrows.



WSTOEN = 3, WST1 = 6

Fig 5. Reading from external memory

A timing diagram for writing to external memory is shown In Figure 6. The relationship between wait-state settings is indicated with arrows.



WSTWEN = 3, WST2 = 7

(1) $\overline{BLS}$ has the same timing as $\overline{WE}$ in configurations that use the byte lane enable signals to connect to write enable (8 bit devices).

Fig 6. Writing to external memory