8-bit LGT8XM

RISC Microcontroller with
In-System Programmable
FLASH Memory

LGT8F88P

LGT8F168P

LGT8F328P

Data book

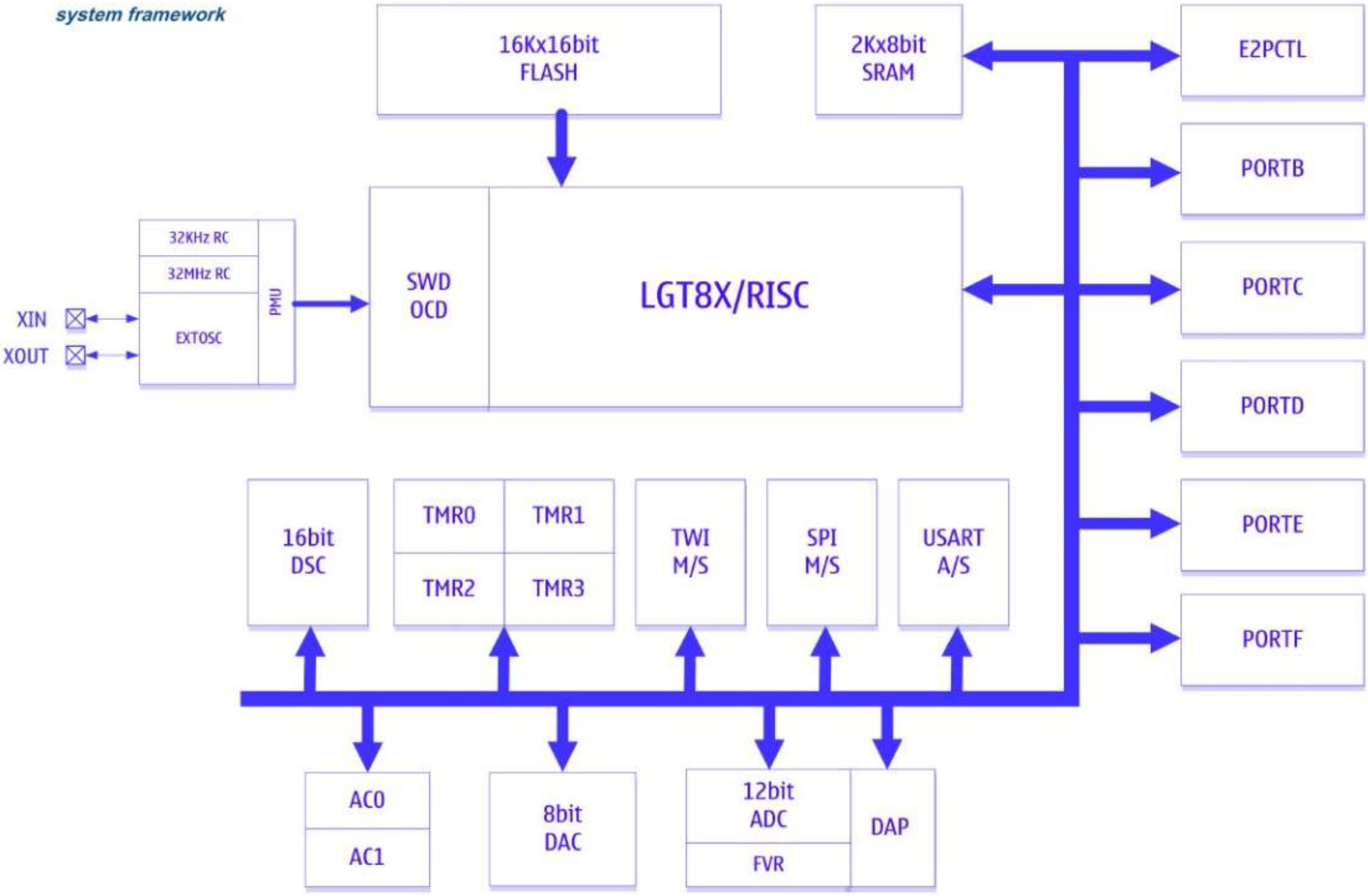
Version 1.0.4J

This is the final list of new Registers that the AVR does not have.

Addr	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0xF6	GUID3	GUID Byte 3							
0xF5	GUID2	GUID Byte 2							
0xF4	GUID1	GUID Byte 1							
0xF3	GUID0	GUID Byte 0							
0xF2	PMCR	PMCE	CLKFS	CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN
0xF0	PMX2	WCE	STOSC1	STOSC0	-	-	XIEN	E6EN	C6EN
0xEE	PMX0	PMXCE	C1BF4	C1AF5	C0BF3	C0AC0	SSB1	TXD6	RXD5
0xED	PMX1	-	-	-	-	-	C3AC	C2BF7	C2AF6
0xEC	TCKCSR	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
0xE2	PSSR	PSS1	PSS3	-	-	-	-	PSR3	PSR1
0xE1	OCPUE	PUE7	PUE6	PUE5	PUE4	PUE3	PUE2	PUE1	PUE0
0xE0	HDR	-	-	HDR5	HDR4	HDR3	HDR2	HDR1	HDR0
0xDE	DAPTE	DAPTE	-	-	-	-	-	-	-
0xDD	DAPTR	DAPTP			DAP Trimming				
0xDC	DAPCR	DAPEN	GA1	GA0	DNS2	DNS1	DNS0	DPS1	DPS0
0xCF	LDOCR	WCE				PDEN	VSEL2	VSEL1	VSEL0
0xCE	VCAL2	Calibration value for 2.048V internal reference							
0xCD	VCAL1	Calibration value for 1.024V internal reference							
0xCC	VCAL3	Calibration value for 4.096V internal reference							
0xC8	VCAL	Internal Voltage Reference calibration register							
0xAF	DPS2R	-	-	-	-	DPS2E	LPRCE	TOS1	TOS0
0xAE	IOCWK	IOCD7	IOCD6	IOCD5	IOCD4	IOCD3	IOCD2	IOCD1	IOCD0
0xAD	ADCSRD	BGEN	REFS2	IVSEL1	IVSEL0	-	VDS2	VDS1	VDS0
0xAC	ADMSC	AMOF	-	-	-	AMFC3	AMFC2	AMFC1	AMFC0
0xAB	ADT1H	ADC Auto-monitor Overflow threshold high byte							
0xAA	ADT1L	ADC Auto-monitor Overflow threshold low byte							
0xA9	PORTE	Port Output E (for compatible with LGT8FX8D)							
0xA8	DDRE	Data Direction E (for compatible with LGT8FX8D)							
0xA7	PINE	Port Input E (for compatible with LGT8FX8D)							
0xA6	ADT0H	ADC Auto-monitor Underflow threshold high byte							
0xA5	ADT0L	ADC Auto-monitor Underflow threshold low byte							
0xA4	OFR1	ADC positive offset trimming							
0xA3	OFR0	ADC negative offset trimming							
0xA1	DALR	DAC data register							
0xA0	DACON	-	-	-	-	DACEN	DAOE	DAVS1	DAVS0
0x9F	OCR3CH	Compare output register high byte of Timer3 C channel							
0x9E	OCR3CL	Compare output register low byte of Timer3 C channel							
0x9D	DTR3H	Dead-band register high byte of Timer3							
0x9C	DTR3L	Dead-band register low byte of Timer3							
0x9B	OCR3BH	Compare output register high byte of Timer3 B channel							
0x9A	OCR3BL	Compare output register low byte of Timer3 B channel							
0x99	OCR3AH	Compare output register high byte of Timer3 A channel							
0x98	OCR3AL	Compare output register low byte of Timer3 A channel							
0x97	ICR3H	Input capture register high byte of Timer3							
0x96	ICR3L	Input capture register low byte of Timer3							
0x95	TCNT3H	Counter register high byte of Timer3							
0x94	TCNT3L	Counter register low byte of Timer3							
0x93	TCCR3D	Control register D of Timer3							
0x92	TCCR3C	Control register C of Timer3							
0x91	TCCR3B	Control register B of Timer3							
0x90	TCCR3A	Control register A of Timer3							
0x8D	DTR1H	Dead-band register high byte of Timer1							
0x8C	DTR1L	Dead-band register low byte of Timer1							
0x83	TCCR1D	DSX17	DSX16	DSX15	DAX14	-	-	DSX11	DSX10
0x7D	ADCSRC	OFEN	-	SPN	AMEN	-	SPD	DIFS	ADTM
0x76	DIDR2	-	PB5D	-	-	-	-	-	-
0x75	IVBASE	Interrupt Vector Base Address							
0x74	PCMSK4								
0x73	PCMSK3	PCINT[39:32]							
0x71	TIMSK3			ICIE3	-	OCIE3C	OCIE3B	OCIE3A	TOIE3
0x67	RCKCAL	RC32K Calibration							
0x65	PRR1	-	-	PRWDT	-	PRTIM3	PREFL	PRPCI	-
0x62	VDTCR	WCE	SWR	-		VDTS		VDREN	VDTEN
0x5C	E2PD3	E2PCTL Data register byte 3							
0x5B	C1TR	AC1 trimming data							
0x5A	E2PD1	E2PCTL Data register byte1							
0x59	DSAH	DSA[31:16] access port of uDSC							
0x58	DSAL	DSA[15:0] access port of uDSC							
0x56	ECCR	WEN	EEN	ERN	SWM	CP1	CP0	ECS1	ECS0
0x52	C0TR	AC0 Trimming register							
0x51	C0XR	-	C0OE	C0HYSE	C0PS0	C0WKE	C0FEN	C0FS1	C0FS0
0x4F	DTR0	TC0 Dead-band timing control register							
0x49	TCCR0C	DSX07	DSX06	DSX05	DSX04	-	-	DSX01	DSX00
0x3A	C1XR	-	C1OE	C1HYSE	C1PS0	C1WKE	C1FEN	C1FS1	C1FS0
0x39	SPFR	RDFULL	RDEMPT	RDPTR1	RDPTR0	WRFULL	WREMPT	WRPTR1	WRPTR0
0x38	TIFR3	-	-	ICF3	-	-	OCF3B	OCF3A	TOV3
0x34	PORTF	Port Output of Group F							
0x33	DDRF	Data Direction of Group F							
0x32	PINF	Port Input of Group F							
0x31	DSDY	DSDY access port of uDSC							
0x30	DSDX	DSDX access port of uDSC							
0x2F	C1SR	C1D	C1BG	C1O	C1I	C1IE	C1IC	C1IS	
0x2E	PORTE	Port Output of Group E							
0x2D	DDRE	Data Direction of Group E							
0x2C	PINE	Port Input of Group E							
0x22	DSSD	DSSD access port of uDSC							
0x21	DSIR	Instruction regiter of uDSC							
0x20	DSCR	DSUEN	MM	D1	D0	-	DSN	DSZ	DSC

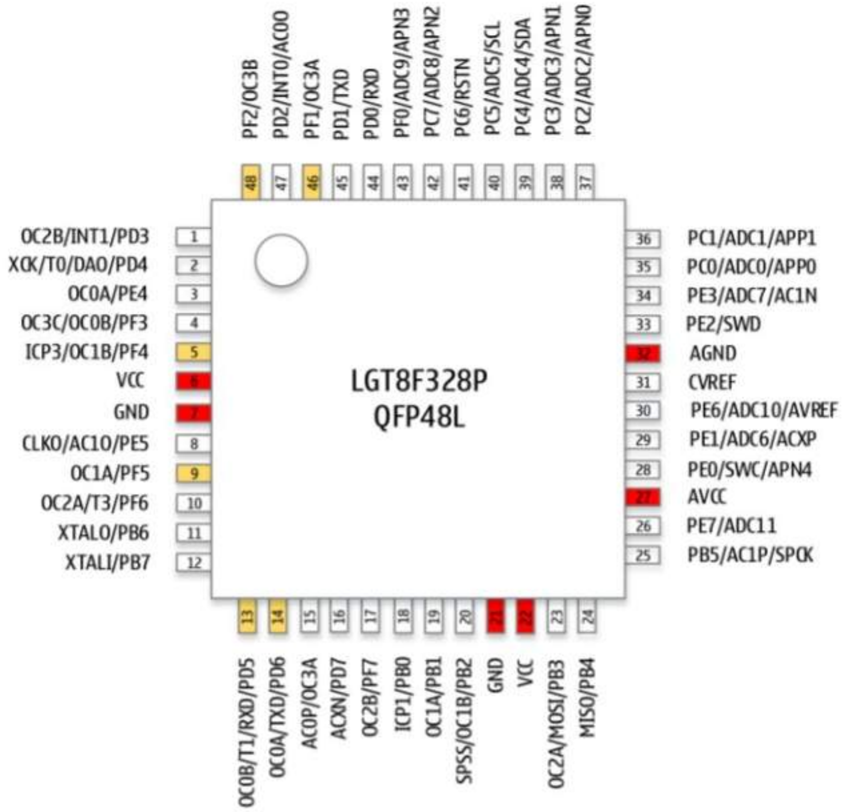
Note: This document has been created for personal use. It is a combination of sources and information. This is not official documentation and is completely unaffiliated with any distribution company or manufacturer. This document is shared freely along with it's source on github. This document is provided as-is in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. In no event will the authors and/or contributors be held liable for any damages arising from the use of this document.

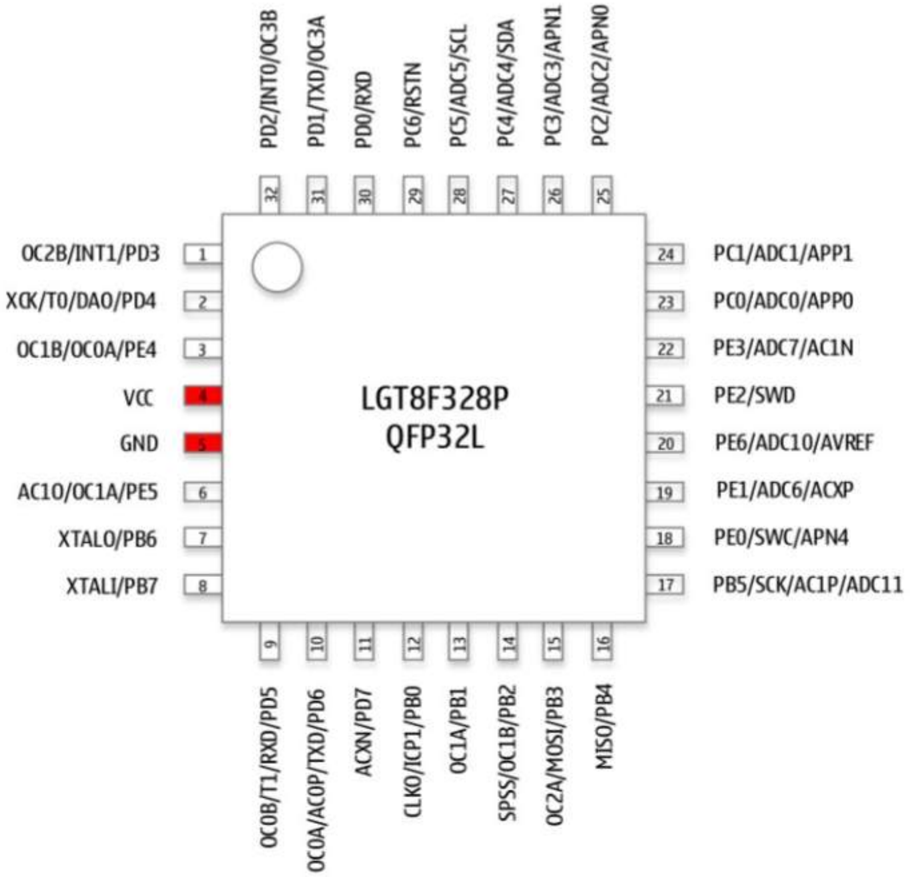
system framework

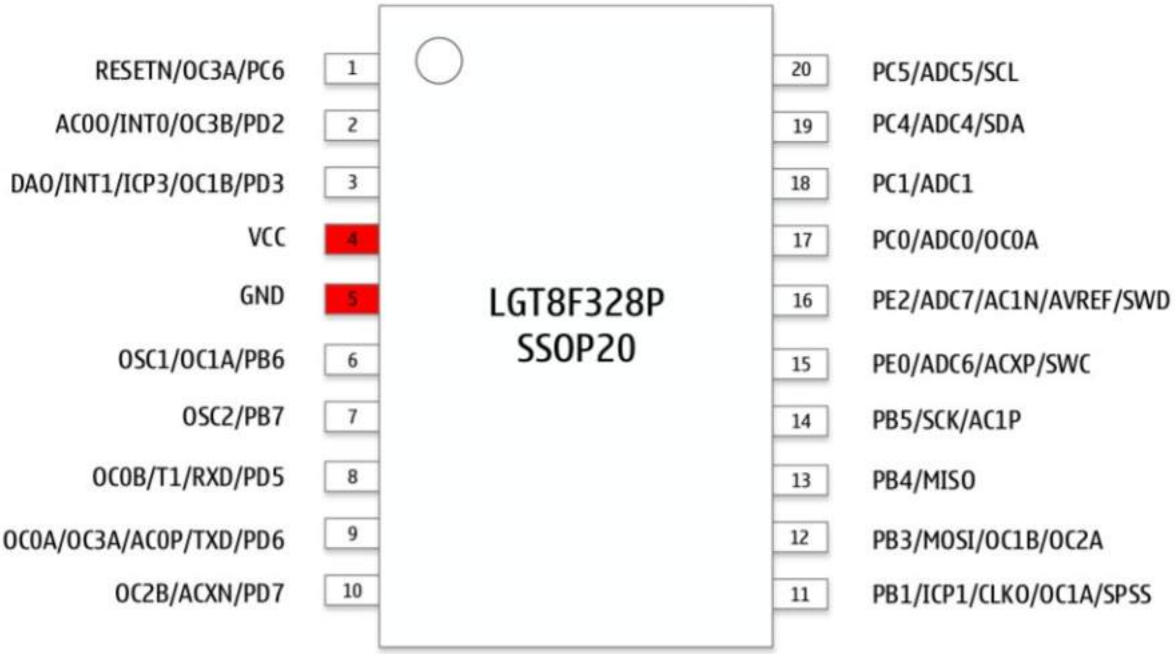


Module Name	Module features
SWD	Debug module, debugging and ISP Features
LGT8X	8Bit high performance RISC Kernel
E2PCTL	FLASH Data Access Interface Controller
PMU	Power management module
PORTB/C/D/E/F	General-purpose programmable input and output ports
DSC	16 Digit arithmetic acceleration unit
ADC	8 Channel 12 Bit ADC with programmable gain
DAP	Differential amplifier
IVREF	1.024V / 2.048V / 4.096V Internal Reference
ACO/1	Analog comparator
TMRO/ 1/2/3	8/16 Bit timer / counter, PWM Controller
WDT	Reset Watchdog module
SPIM/S	Master-slave SPI Controller
TWIM/S	Master-Slave two-wire interface controller, compatible I2C protocol
USART	Synchronous / Asynchronous Serial Transceiver
DAC	8 Bit DAC

Package defined






RESETN/OC3A/PC6
1
ACO0/INT0/OC3B/PD2
2
DAO/INT1/ICP3/OC1B/PD3
3
VCC
4
GND
5
OSC1/OC1A/PB6
6
OSC2/PB7
7
OC0B/T1/RXD/PD5
8
OC0A/OC3A/AC0P/TXD/PD6
9
OC2B/ACXN/PD7
10
LGT8F328P
SSOP20
20
PC5/ADC5/SCL
19
PC4/ADC4/SDA
18
PC1/ADC1
17
PC0/ADC0/OC0A
16
PE2/ADC7/AC1N/AVREF/SWD
15
PE0/ADC6/ACXP/SWC
14
PB5/SCK/AC1P
13
PB4/MISO
12
PB3/MOSI/OC1B/OC2A
11
PB1/ICP1/CLKO/OC1A/SPSS

Pin Description

QFP48	QFP32	SSOP20	
01	01	03	PD3/INT1/OC2B*
			PD3: I/O Pin D3 INT1: External Interrupt 1 Input OC2B: Timer/Counter 2 Output Compare Match Output B
02	02		PD4/DAO/T0/XCK
			PD4: I/O Pin D4 DAO: Digital to Analog Converter DAC T0: Timer0 Timer/Counter 0 XCK: USART USART external clock
03	03	-	PE4/OC0A*
			PE4: I/O Pin E4 OC0A: Timer/Counter 0 Output Compare Match Output A
04	-	-	PF3/OC3C/OC0B*
			PF3: I/O Pin F3 OC3C: Timer/Counter 3 Output Compare Match Output C OC0B: Timer/Counter 0 Output Compare Match Output B
05	03	03	PF4/OC1B*/ICP3
			PF4: I/O Pin F4 OC1B: Timer/Counter 1 Output Compare Match Output B ICP3: Timer 3 Capture Input
06	04	04	VCC
07	05	05	GND
08	06	-	PE5/AC10/CLKO*
			PE5: I/O Pin E5 C10: Comparator AC1 Output CLKO: System Clock Output
09		06	PF5/OC1A*
			PF5: I/O Pin F5 OC1A: Timer/Counter 1 Output Compare Match Output A
10	-	-	PF6/T3/OC2A*
			PF6: I/O Pin F6 T3: Timer 3 External clock input OC2A: Timer/Counter 2 Output Compare Match Output A
11	07	06	PB6/XTALO
			PB6: I/O Pin B6 XTALO: Crystal Oscillator Output

QFP48	QFP32	SSOP20	
12	08	07	PB7/XTALI PB7: I/O Pin B7 XTALI: Crystal Oscillator Input
13	09	08	PD5/RXD*/T1/OC0B PD5: I/O Pin D5 RXD: USART Receive Data T1: Timer 1 External Clock Input OC0B: Timer/Counter 0 Output Compare Match B
14	10	09	PD6/TXD*/OC0A PD6: I/O Pin D6 TXD: USART Transmit Data OC0A: Timer/Counter 0 Output Compare Match A
15			AC0P/OC3A AC0P: Analog Comparator 0 Positive Input OC3A: Timer/Counter 3 Output Compare Match A
16	11	10	PD7/ACXN PD7: I/O Pin D7 ACXN: Analog Comparator 0/1 Inverting Input
17	-		PF7/OC2B PF7: I/O Pin F7 OC2B: Timer/Counter 2 Output Compare Match B
18	12	11	PB0/ICP1 PB0: I/O Pin B0 ICP1: Timer 1 Capture Input
19	13		PB1/OC1A PB1: I/O Pin B1 OC1A: Timer/Counter 1 Output Compare Match A
20	14	12	PB2/OC1B/SPSS PB2: I/O Pin B2 OC1B: Timer/Counter 1 Output Compare Match B SPSS: SPI Slave Select
21	-	-	GND
22	-	-	VCC
23	15	12	PB3/MOSI/OC2A PB3: I/O Pin B3 MOSI: SPI Master Output Slave Input OC2A: Timer/Counter 2 Output Compare Match A
24	16	13	PB4/MISO PB4: I/O Pin B4 MISO: SPI Master Input Slave Output
25	17	14	PB5/SPCK/AC1P PB5: I/O Pin B5 SPCK: SPI Clock AC1P: Analog Comparator 1 Noninverting Input
QFP48	QFP32	SSOP20	

QFP48	QFP32	SSOP20	
26	-	-	PE7/ADC11 PE7: I/O Pin E7 ADC11: ADC Input Channel 11
27	-	-	AVCC: Internal Analog Circuit Positive Power Supply
28	18	15	PE0/SWC/APN4 PE0: I/O Pin E0 SWC: SWD Debug Interface Clock APN4: Differential Amplifier Inverting Input Channel 4
29	19		PE1/ADC6/ACXP PE1: I/O Pin E1 ADC6: ADC Input Channel 6 ACXP: Analog Comparator 0/1 Noninverting Input
30	20	16	PE6/ADC10/AVREF PE6: I/O Pin E6 ADC10: ADC Input Channel 10 AVREF: ADC External Reference Voltage Input
31	-	-	CVREF: ADC Reference Voltage External Filter Capacitor (0.1uF)
32	-	-	AGND: Internal Analog Circuit Power Supply Ground
33	21	16	PE2/SWD PE2: I/O Pin E2 SWD: SWD Debug Interface Data
34	22		PE3/ADC7/AC1N PE3: I/O Pin E3 ADC7: ADC Input Channel 7 AC1N: Analog Comparator 1 Inverting Input
35	23	17	PC0/ADC0/APP0 PC0: I/O Pin C0 ADC0: ADC Input Channel 0 APP0: Differential Amplifier Channel 0 Positive Input
36	24	18	PC1/ADC1/APP1 PC1: I/O Pin C1 ADC1: ADC Input Channel 1 APP1: Differential Amplifier Channel 1 Positive Input
37	25	-	PC2/ADC2/APN0 PC2: I/O Pin C2 ADC2: ADC Input Channel 2 APN0: Differential Amplifier Channel 0 Inverting Input
38	26	-	PC3/ADC3/APN1 PC3: I/O Pin C3 ADC3: ADC Input Channel 3 APN1: Differential Amplifier Channel 1 Inverting Input
QFP48	QFP32	SSOP20	

QFP48	QFP32	SSOP20	
39	27	19	PC4/ADC4/SDA
			PC4: I/O Pin C4
			ADC4: ADC Input Channel 4
			SDA: I2C Data
40	28	20	PC5/ADC5/SCL
			PC5: I/O Pin C5
			ADC5: ADC Input Channel 5
			SCL: I2C Clock
41	29	1	PC6/RESETN
			PC6: I/O Pin C6
			RESETN: External Reset Input
42	-	-	PC7/ADC8/APN2
			PC7: I/O Pin C7
			ADC8: ADC Input Channel 8
			APN2: Differential Amplifier Channel 2 Inverting Input
43	-	-	PF0/ADC9/APN3
			PF0: I/O Pin F0
			ADC9: ADC Input Channel 9
			APN3: Differential Amplifier Channel 3 Inverting Input
44	30	-	PD0/RXD
			PD0: I/O Pin D0
			RXD: USART Receive
45	31	-	PD1/TXD
			PD1: I/O Pin D1
			TXD: USART Transmit
46		1	PF1/OC3A
			PF1: I/O Pin F1
			OC3A: Timer/Counter 3 Output Compare Match A
47	32	2	PD2/INT0/AC00
			PD2: I/O Pin D2
			INT0: External Interupt 0 Input
			AC00: Analog Comparator 0 Output
48			PF2/OC3B
			PF2: I/O Pin F2
			OC3B: Timer/Counter 3 Output Compare Match B
QFP48	QFP32	SSOP20	

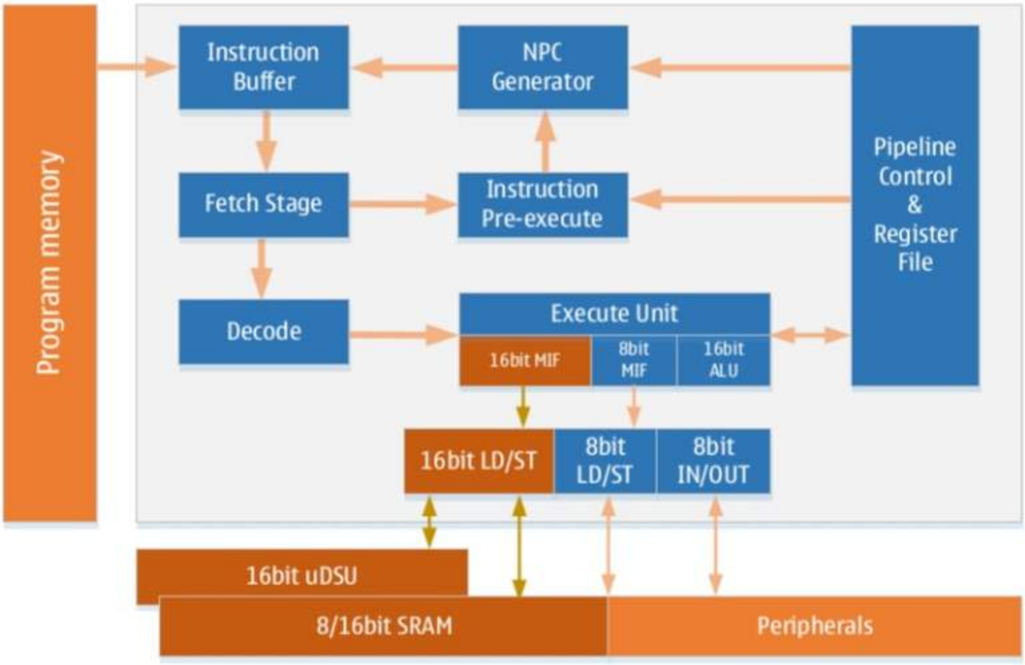
LGT8XM

- Low Power Design
- High Efficency RISC Archatecture
- 16 Bit LD/ST Extension (Dedicated uDSU)
- 130 Instructions, Over 80% execute in a Single Cycle
- Embedded In-Circuit Debugger (OCD)

OVERVIEW

This section discusses the LGT8XM core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

LGT8XM



In order to maximize performance and parallelism, the LGT8XM uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is prefetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory. The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

The thirty-two extra working registers are used to Form a combination of three 16-bit registers that can be used to indirectly address the address pointer for accessing external memory and between FLASH programs. The LGT8XM supports single-cycle 16-bit arithmetic, greatly improving the efficiency of indirect addressing. The three special 16-bit registers in the LGT8XM core are named X,Y, Z registers, which are described in more detail later.

The ALU supports arithmetic logic operations between registers and between constants and registers. The operation of a single register can also be performed in the ALU. After the ALU operation is completed, the effect of the operation on the state of the kernal is updated to the status register (SREG). Program Flow control can be addressed to the program area by conditional and unconditional jump/call implementations. Most LGT8XM instructions are 16 bits. Each program address space correspondsto a 16-bit or 32-bit LGT8XM instruction.

After the kernel responds to an interrupt or a subroutine call, the return address (PC) is stored on the stack. The stack is allocated in the system's general data SRAM, so the size of the stack is limited only by the size and usageof the system's SRAM. All applications that support interrupt or subroutine calls must first initialize the Stack Pointer Register (SP), which can be accessed through the IO space. Data SRAM can be accessed in Five different addressing modes. The internal memory of the LGT8XM is linearly mapped to a uniform address space. Please refer to the introduction of the storage memory chapter for further details.

The LGT8XM core includes a Flexible interrupt controller that can be controlled by a global interrupt enable bit in the status register. All interrupts have a separate interrupt vector. The priority of the interrupt has a corresponding relationship with the interrupt vector address. The smaller the interrupt address, the higher the priority of the interrupt. The I/O space contains 64 register spaces that can be directly addressed by IN/OUT instructions. These registers are used for kernel control as well as control Functions For status registers, SPI and other I/O peripherals. This space can be accessed directly by the IN/OUTinstruction or by their address mapped to the data memory space (0x20 - Ox5F). In addition, the LGT8FX8P also includes extended I/O space, which is mapped to data memory space 0x60 — OxFF, which can only be accessed using ST/STS/STD and LD/LDS/LDD instructions.

To enhance the computing power of the LGT8XM core, a 16-bit LD/ST extension has been added to the instruction set. This 16-bit LD/ST expansion works with the 16-Dimensional Operation Acceleration Unit (UDSU) For efficient 16-bit data operations. At the same time, the kernel also increases the 16-bit access capability to the RAM space. So the 16-bit LD/ST extension can pass 16 bits of data between the uDSU, RAM, and working registers. Please refer to the "Digital Operation Accelerator" section For details.

ARITHMETIC LOGIC UNIT (ALU)

The LGT8XM internally contains a 16-bit arithmetic logic unit that can perform 16 bit arithmetic operations on data in one cycle. The highly efficient ALU is connected to 32 general purpose working registers. The ALU has the ability to perform two arithmetic operations between registers or registers and immediate data in one cycle. There are three types of ALU operations:arithmetic, logic, and bit operations. The ALU also includes a single-cycle hardware multiplier that implements direct signed or unsigned operations on two 8-bit registers in a single cycle. Please refer to the detailed description in the instruction set section.

STATUS REGISTER (SREG)

The status register mainly stores the result information generated by the execution of the most recent ALU operation. This information is used to control the program execution Flow. The status register is updated after the ALU operation has completely ended, thus eliminating the need for separate compare instructions, resulting in a more compact and efficient code implementation. The value of the status register is not automatically saved and restored in response to an interrupt and exit from the interrupt, which requires software to implement.

SREG *REGISTER DEFINITION*

<table>
<tr><th colspan="9"><b>SREG</b> System Status Register</th></tr>
<tr><td colspan="4">Address: <b>0X3F (0X5F)</b></td><td colspan="5">Defaults: <b>0X00</b></td></tr>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Name</td><td>I</td><td>T</td><td>H</td><td>S</td><td>V</td><td>N</td><td>Z</td><td>C</td></tr>
<tr><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
<tr><td colspan="9">Bit Definition</td></tr>
<tr><td>[0]</td><td>C</td><td colspan="7">Carry flag, indicating that the arithmetic or logic operation caused the carry. *</td></tr>
<tr><td>[1]</td><td>Z</td><td colspan="7">A zero flag indicating that the result of an arithmetic or logical operation is zero. *</td></tr>
<tr><td>[2]</td><td>N</td><td colspan="7">A negative flag indicates that an arithmetic or logic operation has produced a negative number. *</td></tr>
<tr><td>[3]</td><td>V</td><td colspan="7">The overflow flag indicates that the result of the two's complement operation has overflowed. *</td></tr>
<tr><td>[4]</td><td>S</td><td colspan="7">Sign bit, equivalent to the XOR operation result of N and V *</td></tr>
<tr><td>[5]</td><td>H</td><td colspan="7">Semi-carry flag, useful in BCD operations, indicating the semi-carry generated by byte operations*</td></tr>
<tr><td>[6]</td><td>T</td><td colspan="7">For temporary bits, bit copy (BLD) and bit store (BST) instructions, the T bit is used as a temporary memory bit to temporarily store the value of a bit in the general purpose register. *</td></tr>
<tr><td>[7]</td><td>I</td><td colspan="7">The global interrupt enable bit must be set to 1 to enable the core to respond to interrupt events. Different interrupt sources are controlled by independent control bits. The global interrupt enable bit is the last barrier that controls the interrupt signal into the core. The I bit is automatically cleared by the hardware after the core responds to the interrupt vector and is automatically set after the interrupt return instruction (RETI) is executed. The I bit can also be changed using the SEI and CLI instructions. *</td></tr>
</table>

***** *Please refer to the instruction description for details.*

GENERAL WORKING REGISTER

The general purpose working registers are optimized according to the LGT8XM instruction set architecture. To achieve the efficiency and flexibility required for core execution, the LGT8XM's internal working registers support several access modes:

- An 8-bit read and an 8-bit write simultaneously
- Two 8-bit reads simultaneously with an 8-bit write
- Two 8-bit reads simultaneously one 16-bit write
- One 16-bit read and one 16-bit write simultaneously

LGT8XM *GENERAL WORKING REGISTER*

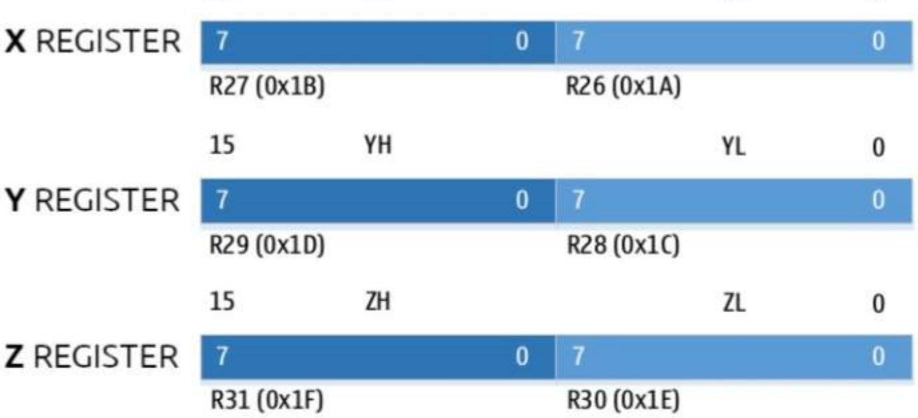
GENERAL WORKING REGISTER

7 0	Addr.	
R0	0x00	
R1	0x01	
R2	0x02	
...		
R13	0x0D	
R14	0x0E	
R15	0x0F	
R16	0x10	
R17	0x11	
...		
R26	0x1A	**X** Register Low Byte
R27	0x1B	**X** Register High Byte
R28	0x1C	**Y** Register Low Byte
R29	0x1D	**Y** Register High Byte
R30	0x1E	**Z** Register Low Byte
R31	0x1F	**Z** Register High Byte

Most of the instructions have direct access to all of the general working registers, and most of them are also single-cycle instructions. As shown in the figure above, each register corresponds to the address of a data storage space, and these general working registers are mapped to the data storage space. These registers only really exist in SRAM, but this unified mapping storage organization gives them a lot of flexibility. The X/Y/Z register can be indexed as a pointer to any general purpose register.

X/Y/Z REGISTER

Registers R26...R31 can be combined in pairs to form three 16-bit registers. These three 16-bit registers are mainly used as address pointers for indirect addressing access. The X/Y/Z registers are structured as follows:



These registers are used as fixed offset, auto-increment, and auto-decrement address pointers in different addressing modes. See the Instruction Description section for details.