DATA SHEET

GPMQ9103A

ARM® Cortex®-M0 32-bit Microcontroller Wireless Power Series

Preliminary

Nov. 17, 2020

Version 0.3

Table of Contents

Page

ARM® Cortex®-M0 32-bit
Microcontroller Wireless Power Series

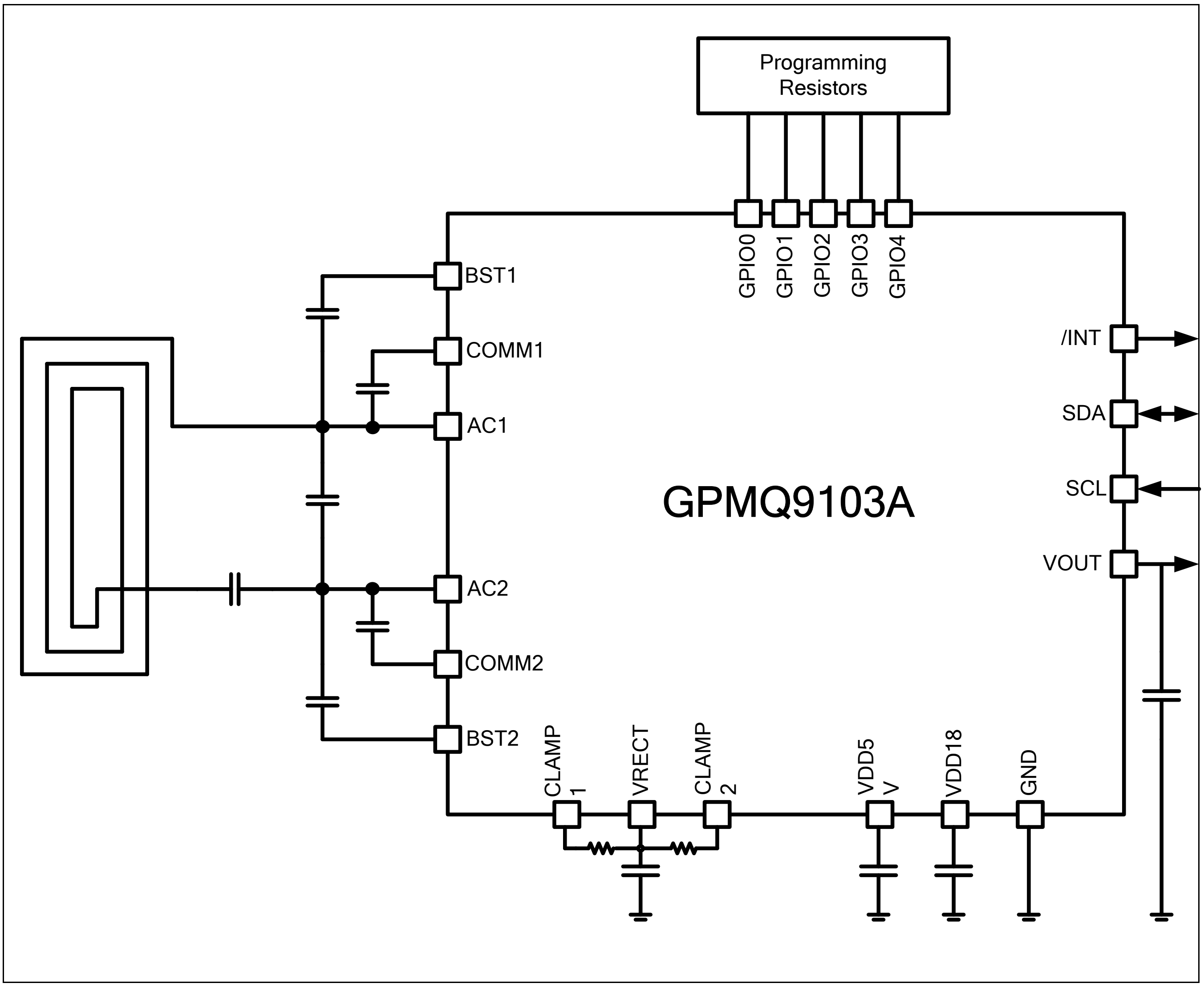
1. GENERAL DESCRIPTION

GPMQ9103A is a high-efficiency, Qi-compliant wireless power receiver targeted for applications up to 15W. Using magnetic inductive charging technology, the receiver converts an AC power signal from a resonant tank to a programmable regulated 5V, 7V, 9V or 12V DC output voltage. The integrated, low RDS_{ON} synchronous rectifier and ultra-low dropout linear (LDO) regulator offer high efficiency, making the product ideally suited for battery-operated applications. ***GPMQ9103A*** is available in a 52-WLCSP package, and it rated for a 0 to 85°C ambient operating temperature range.

2. FEATURES

- **WPC compliant**
  - Single-chip wireless power receiver up to 15W application
  - WPC Qi compliant
- **Synchronous Rectifier**
  - 96% Efficiency for Fully Synchronous Rectifier
  - 97% Efficient for Post Regulator
  - 85% System Efficient at 15W with ***GPMQ8006B TX***
  - Integrated Synchronous Full Bridge Rectifier Circuit
    - 3 modes auto-switch
      - **Synchronous rectification**
        (VRECT ≧ UVLO, dynamical switched by IOUT > *ITH)*
      - **Diode rectification**
        (VRECT ≧ UVLO, dynamical switched by IOUT < *ITH)*
      - **Diode bridge**
        (VRECT < UVLO)
    - Low R_{DS} ON resistance
      - Hi-side 35mΩ
      - Low-side 35mΩ
    - Under Voltage Lockout (UVLO)
      - Threshold volt : 2.8v detect on VRECT
  - Output Voltage
    - **LDO Mode**
      - Configurable VOUT Level: 5v, 7v, 9v, 12v
      - Configurable VOUT Deviation: -2%, 0%, 2%, 4%
    - **Switch Mode**
      - VOUT follow up with VRECT
    - **OFF Mode (VOUT floating)**
      - External EN pin = 1
      - UVLO occurs
      - Register VOUT_EN = 0
      - OTP occurs.
  - Maximum output current : 1.67A (OCL = 2A)
- **System Protection**
  - VRECT Over Voltage Protection (Detection)
    - Configurable detected level: 15V, 17V, 19V, 21V
  - Over Current Limit
    - Configurable 1A , 1.5A, 2A
  - Over Temperature Protection
    - 140˚C, thermal shutdown
    - 130˚C, thermal warning
  - VOUT Over Voltage Protection
    - Detected level as VOUT *1.25
- **FSK De-modulator**
  - AC1 FSK de-modulated
- **CPU Subsystem**
  - CPU Core
    - ARM® Cortex®-M0 32-bit CPU (48MHz max)
- **Memories**
  - 2K Bytes System RAM
  - Up to 16K Bytes One Time Programmable Memory
- **Clock Management**
  - Internal oscillator: 48MHz±1.5% @ 3V~5.5V with -40°C to 85°C
- **Power Management**
  - Power off when External EN pin goes high.
    - Regulator 5V switch to low current mode
    - Regulator 1.8V shut down
- **Reset Management**
  - Power On Reset
    - Power RESET state, VRECT < UVLO
  - PAD Reset (SCL/PADRESETn_RSTn)
  - Master Reset (MRESETnn)
  - System Reset (SRESETnn)
- **Analog peripherals**
  - 10Bit A/D Converters with 107 KSPS and 12 analog inputs
    - Reference voltage as 2.1v
    - User-programmable foreign-object detection (FOD)
    - Dedicated remote temperature sensing
  - Regulator 5V
  - Regulator 1.8V
- **System Control**
  - Watchdog Timer (WDT) for safety sensitive applications
  - System Management Unit (SMU) for system configuration and control

- **I/O Ports**
  - External enable active low, input pin.
  - Interrupt pin (INT), open drain pin.
  - Max 5 multifunction bi-direction I/Os
- **Timer**
  - Up to 3 general timers with 16-bit counter
- **Communication peripherals**
  - Built-in I2C
- **Debug System**
  - ARM serial wire debug (SWD)
  - Supporting up to 4 hardware breakpoints
  - Supporting up to two watch points
- **Operation Temperature**
  - -40°C to 85°C
- **Reliability**
  - HBM 2KV
  - MM 200V
  - CDM 500V
- **Package**
  - WLCSP 2.64X3.94-52B
  - TQFN5x5-40



3. FAMILY DEVICE COLLECTIONS

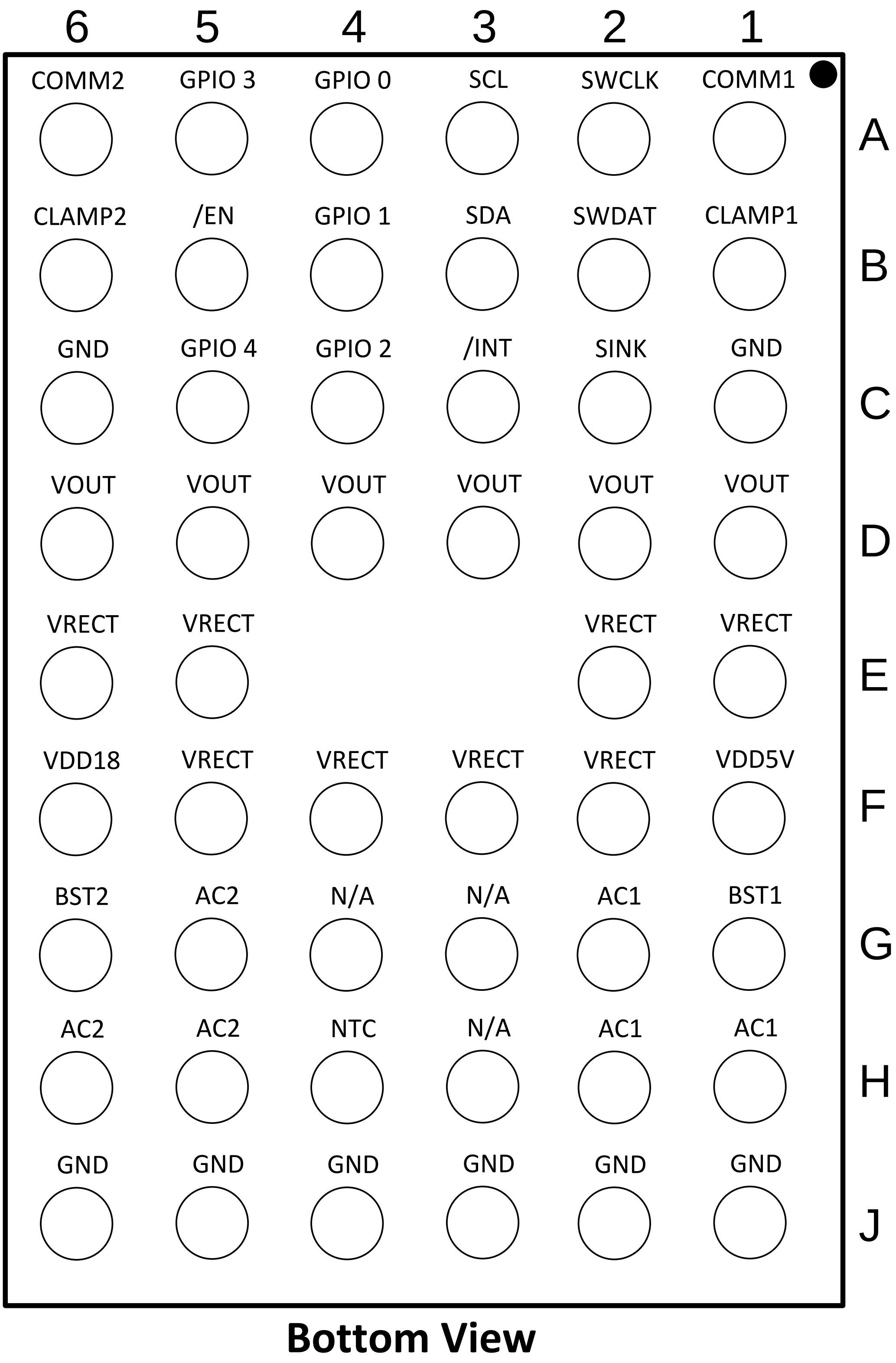
The following table lists the available features per device type for the GPMQ9103A series.

GPMQ9103A Base Series Selection Guide

Features		**GPMQ9103A**	**GPMQ9103A**
CPU frequency		48MHz	48MHz
Operating temperature		Ambient temperature -40 to 85 °C	Ambient temperature -40 to 85 °C
Operating voltage		3v to 5.5v	3v to 5.5v
OTP Memory (Kbytes)		16	16
SRAM (Kbytes)		2	2
Industrial Control	CCU4	--	--
	ADC CTRL	1	1
	TIMER/CAPT	4	4
Communication	UART	--	--
	I^2C	1	1
	SPI	--	--
Analog	ADC	1	1
	Temperature Sensor	1	1
	LVR	1	1
	IOSC48M	1	1
	Synchronous Full Bridge Rectifier	1	1
	HV Power LDO	1	1
	FSK De-modulator	1	1
I/Os		5	5
Package		TQFN5x5-40	WLCSP 2.64X3.94-52B

Package Pinouts

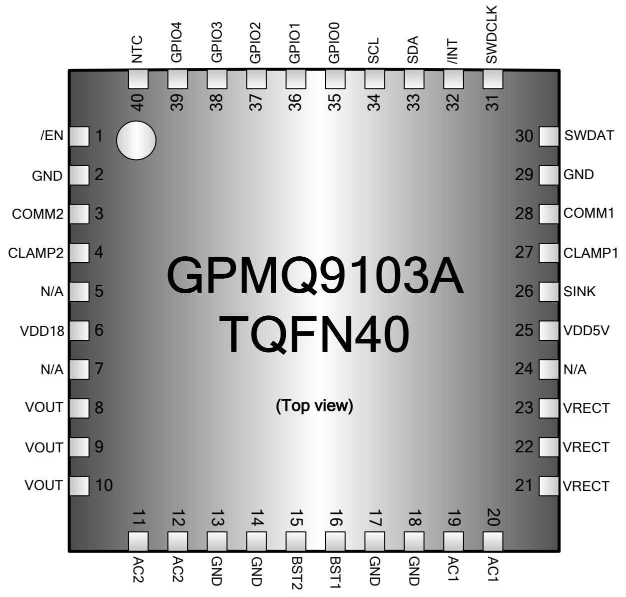
WLCSP package pinout



	6	5	4	3	2	1
A	COMM2	GPIO 3	GPIO 0	SCL	SWCLK	COMM1
B	CLAMP2	/EN	GPIO 1	SDA	SWDAT	CLAMP1
C	GND	GPIO 4	GPIO 2	/INT	SINK	GND
D	VOUT	VOUT	VOUT	VOUT	VOUT	VOUT
E	VRECT	VRECT			VRECT	VRECT
F	VDD18	VRECT	VRECT	VRECT	VRECT	VDD5V
G	BST2	AC2	N/A	N/A	AC1	BST1
H	AC2	AC2	NTC	N/A	AC1	AC1
J	GND	GND	GND	GND	GND	GND

Bottom View

TQFN40 (5x5) package pinout



Pin description

ALT* = Alternate Functions Mode

TQFN40 PIN	WLCSP PIN	NAME	TYPE	Function
28	A1	COMM1	Output (Open-drain)	Open-drain output used to communicate with the transmitter.
31	A2	RSTn / SWCLK	Input	Reset pin / Writer Clock / Serial Wire Debug Clock.
34	A3	SCL	Bi-I/O	I2c Serial clock line. Open-drain pin.
35	A4	GPIO 0	Bi-I/O	General purpose I/O & configurable as ADC input channel 6
38	A5	GPIO 3	Bi-I/O	General purpose I/O & configurable as ADC input channel 9
3	A6	COMM2	Output (Open-drain)	Open-drain output used to communicate with the transmitter.
27	B1	CLAMP1	Output (Open-drain)	Open-drain output used to overvoltage protection on the AC side.
30	B2	SWDAT	Input Bi-I/O	Writer Data / Serial Wire Debug Data
33	B3	SDA	Input/Output (Open-drain)	I2c Serial data line. Open-drain pin.
36	B4	GPIO 1	Bi-I/O	General purpose I/O & configurable as ADC input channel 7
1	B5	/EN	Input	Active-LOW enable pin. Pulling this pin to logic HIGH forces the device into Shut Down Mode. When connected to logic LOW, the device is enabled. Do not leave this pin floating.
4	B6	CLAMP2	Output (Open-drain)	Open-drain output used to overvoltage protection on the AC side.
2,13,14,	C1, C6, J1,	GND	GND	Ground.

TQFN40 PIN	WLCSP PIN	NAME	TYPE	Function
17,18,29	J2,J3, J4, J5, J6			
26	C2	SINK	Output (Open-drain)	Open-drain output for controlling the rectifier clamp. Connect with a resistor from this pin to the VRECT pin.
32	C3	/INT	Output (Open-drain)	Interrupt flag pin. This is an open-drain output that signals fault interrupts. It is also asserted LOW when /EN is HIGH. Connect /INT to VDD18 through a resistor.
37	C4	GPIO 2	Bi-I/O	General purpose I/O & configurable as ADC input channel 8
39	C5	GPIO 4	Bi-I/O	General purpose I/O & configurable as ADC input channel 10
8,9,10	D1, D2, D3, D4, D5, D6	VOUT	Output	Regulated output voltage pin.
21,22,23	E1, E2, E5, E6, F2, F3, F4, F5	VRECT	Output	Output voltage of the synchronous rectifier bridge. The rectifier voltage dynamically changes as the load changes.
25	F1	VDD5V	Output	Internal 5V regulator output voltage for internal use.
6	F6	VDD18	Output	Internal 1.8V regulator output voltage.
16	G1	BST1	Output	Boost capacitor for driving the high-side switch of the internal rectifier.
19,20	G2, H1, H2,	AC1	Input	AC input power. Connect these pins to the resonant capacitance.
5	G3	N/A	–	Reserved
7	G4	N/A	–	Reserved
11,12	G5, H5, H6	AC2	Input	AC input power. Connect to the Rx coil.
15	G6	BST2	Output	Boost capacitor for driving the high-side switch of the internal rectifier.
24	H3	N/A	–	Reserved
40	H4	NTC	Input	End of Charge (EOC) and remote temperature (TS) sensing for over-temperature shutdown. For remote temperature sensing, connect to the NTC thermistor network. If not used, connect this pin to the VDD18 pin through the 10kΩ resistor.

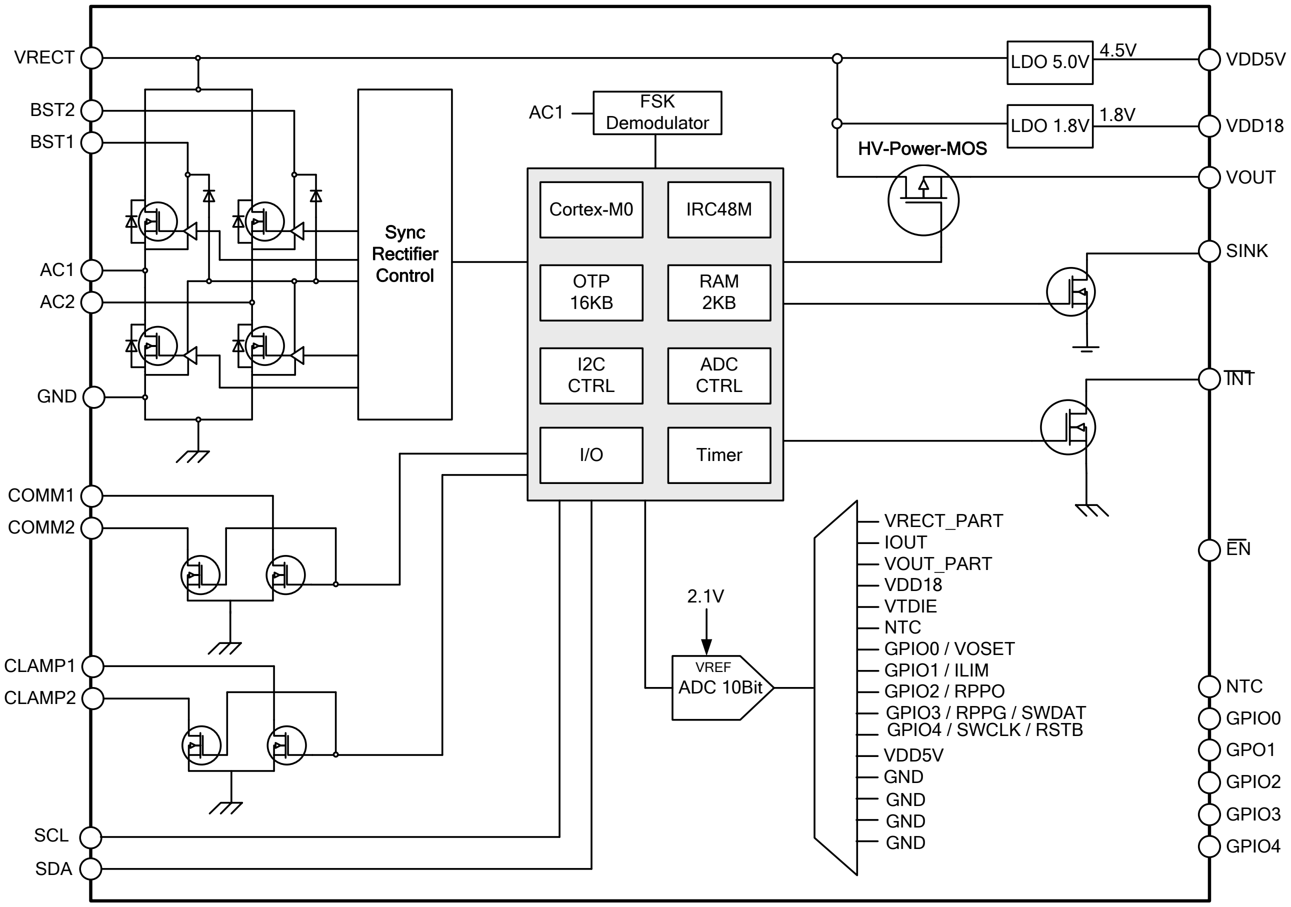
Alternate Functions Map

ALT* = Alternate Functions Mode

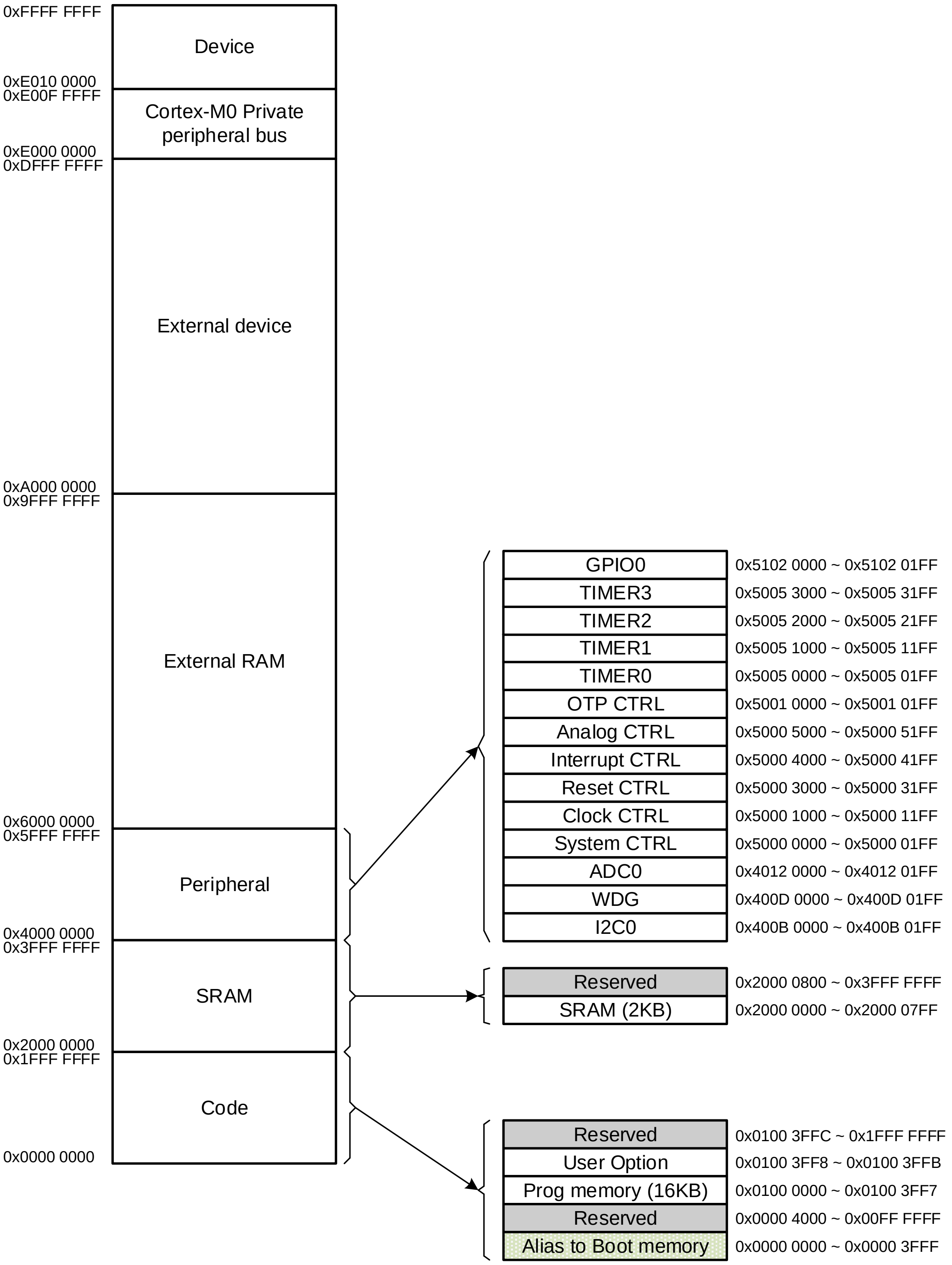
GPMQ9103	PAD TYPE	GPIO ALT_0		ADC ALT_2	
GPIO0	I/O	GP0.0	I/O	ADC_CH6	A
GPIO1	I/O	GP0.1	I/O	ADC_CH7	A
GPIO2	I/O	GP0.2	I/O	ADC_CH8	A
GPIO3	I/O	GP0.3	I/O	ADC_CH9	A
GPIO4	I/O	GP0.4	I/O	ADC_CH10	A
NTC	Input				
SCL	I/O (OD)				
SDA	I/O (OD)				
COMM1	Output (OD)				
COMM2	Output (OD)				
CLAMP1	Output (OD)				
CLAMP2	Output (OD)				
SINK	Output (OD)				
/INT	Output (OD)				
RSTn (SWCLK)	Input				
SWDAT	I/O				
/EN	Input				
AC1	Input				
AC2	Input				
BST1	Output				
BST2	Output				
GND	GND				
VOUT	Output				

4. BLOCK DIAGRAM

4.1. GPMQ9103A



5. MEMORY ORGANIZATION



GPIO0	0x5102 0000 ~ 0x5102 01FF
TIMER3	0x5005 3000 ~ 0x5005 31FF
TIMER2	0x5005 2000 ~ 0x5005 21FF
TIMER1	0x5005 1000 ~ 0x5005 11FF
TIMER0	0x5005 0000 ~ 0x5005 01FF
OTP CTRL	0x5001 0000 ~ 0x5001 01FF
Analog CTRL	0x5000 5000 ~ 0x5000 51FF
Interrupt CTRL	0x5000 4000 ~ 0x5000 41FF
Reset CTRL	0x5000 3000 ~ 0x5000 31FF
Clock CTRL	0x5000 1000 ~ 0x5000 11FF
System CTRL	0x5000 0000 ~ 0x5000 01FF
ADC0	0x4012 0000 ~ 0x4012 01FF
WDG	0x400D 0000 ~ 0x400D 01FF
I2C0	0x400B 0000 ~ 0x400B 01FF

Reserved	0x2000 0800 ~ 0x3FFF FFFF
SRAM (2KB)	0x2000 0000 ~ 0x2000 07FF

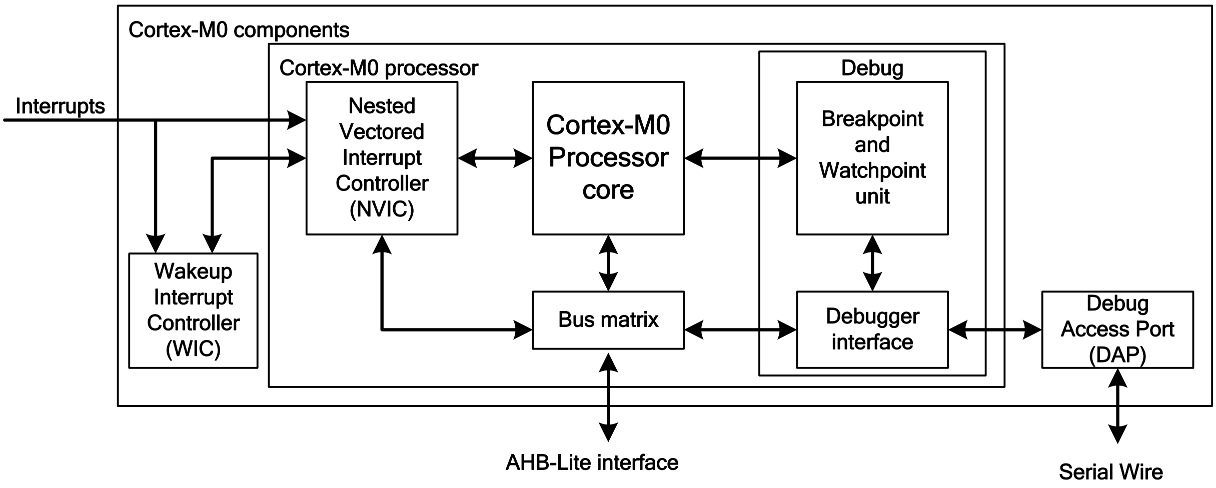
Reserved	0x0100 3FFC ~ 0x1FFF FFFF
User Option	0x0100 3FF8 ~ 0x0100 3FFB
Prog memory (16KB)	0x0100 0000 ~ 0x0100 3FF7
Reserved	0x0000 4000 ~ 0x00FF FFFF
Alias to Boot memory	0x0000 0000 ~ 0x0000 3FFF

6. ARM® Cortex®-M0 Core

6.1. Overview

The Cortex®-M0 processor is an entry-level 32-bit ARM Cortex processor designed for a broad range of embedded applications. It offers significant benefits to developers, including:

- simple, easy-to-use programmers model
- highly efficient ultra-low power operation
- excellent code density
- deterministic, high-performance interrupt handling
- upward compatibility with the rest of the Cortex-M processor family



The Cortex-M0 processor is built on a highly area and power optimized 32-bit processor core, with a 3-stage pipeline von Neumann architecture. The processor delivers exceptional energy efficiency through a small but powerful instruction set and extensively optimized design, providing high-end processing hardware including a single-cycle multiplier. The Cortex-M0 processor implements the ARMv6-M architecture, which is based on the 16-bit Thumb® instruction set and includes Thumb-2 technology. The Cortex-M0 instruction set provides exceptional performance expected of a modern 32-bit architecture, with a higher code density than other 8-bit and 16-bit microcontrollers. The Cortex-M0 processor closely integrates a configurable NVIC, to deliver industry leading interrupt performance. The NVIC provides 4 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of interrupt service routines (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to abandon and restart load-multiple and store-multiple operations. Interrupt handlers do not require any assembler wrapper code, removing any code overhead from the ISRs. Tail-chaining optimization also significantly reduces the overhead when switching from one ISR to another. To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly poared down.

Reference to ARM documentation

- Cortex®-M0 Devices, Generic User Guide
- ARMv6-M Architecture Reference Manual
- Cortex Microcontroller Software Interface Standard (CMSIS)

6.2. System Control Block

The System Control Block (SCB) provides system implementation information, and system control. This includes configuration, control, and reporting of the system exceptions.

6.2.1. System Control Block usage

Ensure software uses aligned 32-bit word size transactions to access all the system control block registers.

Register Map

Base Address : 0xE000_E000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
CPU_ID	CPUID Base Register	0xD00	R	0x410C C200
ICSR	Interrupt Control and State Register	0xD04	R/W	0x0000 0000
AIRCR	Application Interrupt and Reset Control Register	0xD0C	R/W	0xFA05 0000
SCR	System Control Register	0xD10	R	0x0000 0000
CCR	Configuration and Control Register	0xD14	R/W	0x0000 0208
SHPR2	System Handler Priority Register 2	0xD1C	R/W	0x0000 0000
SHPR3	System Handler Priority Register 3	0XD20	R/W	0x0000 0000
SHCSR	System Handler Control and State Register	0xD24	R/W	0x0000 0000

Registers

CPUID Base Register

The CPUID register contains the processor part number, version, and implementation information.

CPU_ID **CPUID Base Register** **Address : 0xE000 ED00**

31	30	29	28	27	26	25	24
IMPLEMENTER							
23	22	21	20	19	18	17	16
VARIANT				ARCHITECTURE			
15	14	13	12	11	10	9	8
PARTNO							
7	6	5	4	3	2	1	0
PARTNO				REVISION			

Bit	Name	Description	Access	Reset value
[31:24]	**IMPLEMENTER**	Implementer code assigned by ARM. ARM == 0x41	R	0x41
[23:20]	**VARIANT**	Reads as 0x0	R	0x0
[19:16]	**ARCHITECTURE**	Reads as 0xC for ARMv6-M parts	R	0xC
[15:4]	**PARTNO**	Reads as 0xC20.	R	0xC20
[3:0]	**REVISION**	Reads as 0x0	R	0x0

Interrupt Control and State Register

The ICSR:

• provides:

– set-pending and clear-pending bits for the PendSV and SysTick exceptions

• indicates:

– the exception number of the exception being processed

– whether there are preempted active exceptions

– the exception number of the highest priority pending exception

– whether any interrupts are pending.

ICSR **Interrupt Control and State Register** **Address : 0xE000 ED04**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
NMIPENDSET	**--**		**PENDSVSET**	**PENDSVCLR**	**PENDSTSET**	**PENDSTCLR**	**--**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISRPREEMPT	**ISRPENDING**	**--**	**VECTPENDING**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VECTPENDING							**VECTACTIVE**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VECTACTIVE							

Bit	Name	Description	Access	Reset value
[31]	**NMIPENDSET**	**NMI Set-Pending Bit** Write: 0 = No effect 1 = Changes NMI exception state to pending Read: 0 = NMI exception not pending 1 = NMI exception pending **Note:** Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.	R/W	0x0
[30:29]	**--**	Reserved	R	0x0
[28]	**PENDSVSET**	**PendSV Set-Pending Bit** Write: 0 = No effect. 1 = Changes PendSV exception state to pending. Read: 0 = PendSV exception is not pending. 1 = PendSV exception is pending. **Note:** Writing 1 to this bit is the only way to set the PendSV exception state to pending.	R/W	0x0
[27]	**PENDSVCLR**	**PendSV Clear-Pending Bit** Write: 0 = No effect	W	0x0

Bit	Name	Description	Access	Reset value
		1 = Removes the pending state from the PendSV exception **Note:** This bit is write-only. To clear the PENDSV bit, you must "write 0 to PENDSVSET and write 1 to PENDSVCLR" at the same time.		
[26]	**PENDSTSET**	**SysTick Exception Set-Pending Bit** Write: 0 = No effect 1 = Changes SysTick exception state to pending Read: 0 = SysTick exception is not pending. 1 = SysTick exception is pending.	R/W	0x0
[25]	**PENDSTCLR**	**SysTick Exception Clear-Pending Bit** Write: 0 = No effect 1 = Removes the pending state from the SysTick exception. **Note:** This bit is write-only. When you want to clear PENDST bit, you must "write 0 to PENDSTSET and write 1 to PENDSTCLR" at the same time.	W	0x0
[24]	**--**	Reserved	R	0x0
[23]	**ISRPREEMPT**	**Interrupt Preemption Bit** If set, a pending exception will be serviced on exit from the debug halt state. This bit is read only.	R	0x0
[22]	**ISRPENDING**	**Interrupt Pending Flag, Excluding NMI And Faults** 0 = Interrupt not pending 1 = Interrupt pending. This bit is read only.	R	0x0
[21]	**--**	Reserved	R	0x0
[20:12]	**VECTPENDING**	**Exception Number Of The Highest Priority Pending Enabled Exception** 0 = No pending exceptions Non-zero = Exception number of the highest priority pending enabled exception. This bit is read only.	R	0x0
[11:9]	**--**	Reserved	R	0x0
[8:0]	**VECTACTIVE**	**Contains The Active Exception Number** 0 = Thread mode. Non-zero = Exception number of the currently active exception. This bit is read only.	R	0x0

Application Interrupt and Reset Control Register

The AIRCR register provides endian status for data accesses and reset control of the system. To write to this register, you must write 0x5FA to the VECTKEY field, otherwise the processor ignores the write.

AIRCR **Application Interrupt and Reset Control Register** **Address : 0xE000 ED0C**

31	30	29	28	27	26	25	24
VECTORKEY							
23	22	21	20	19	18	17	16
VECTORKEY							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--					SYSRESETREQ	VECTCLRACTIVE	--

Bit	Name	Description	Access	Reset value
[31:16]	**VECTORKEY**	**Register Access Key** Write: When writing to this register, the VECTORKEY field need to be set to 0x05FA, otherwise the write operation would be ignored. The VECTORKEY filed is used to prevent accidental write to this register from resetting the system or clearing of the exception status. Read: Read as 0xFA05.	R	0xFA05
[15:3]	--	Reserved	R	0x0
[2]	**SYSRESETREQ**	**System Reset Request** Writing this bit 1 will cause a reset signal to be asserted to the chip to indicate a reset is requested. The bit is a write only bit and self-clears as part of the reset sequence.	R	0x0
[1]	**VECTCLRACTIVE**	**Exception Active Status Clear Bit** Reserved for debug use. When writing to the register, user must write 0 to this bit, otherwise behavior is unpredictable.	R	0x0
[0]	--	Reserved	R	0x0

System Control Register

The SCR controls features of entry to and exit from low power state.

SCR **System Control Register** **Address : 0xE000 ED10**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			SEVONPEND	--	SLEEPDEEP	SLEEPONEXIT	--

Bit	Name	Description	Access	Reset value
[31:5]	--	Reserved	R	0x0
[4]	**SEVONPEND**	**Send Event On Pending Bit** **Note: *GPMQ9103A not support this function. Please keep it in 0.***	R/W	0x0
[3]	--	Reserved	R	0x0
[2]	**SLEEPDEEP**	**Processor Deep Sleep And Sleep Mode Selection** **Note: *GPMQ9103A not support this function. Please keep it in 0.***	R/W	0x0
[1]	**SLEEPONEXIT**	**Sleep-On-Exit Enable** **Note: *GPMQ9103A not support this function. Please keep it in 0.***	R/W	0x0
[0]	--	Reserved	R	0x0

Configuration and Control Register

The CCR is a read-only register and it indicates some aspects of the behavior of the Cortex-M0 processor..

CCR **Configuration and Control Register** **Address : 0xE000 ED14**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--						STKALIGN	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--				UNALIGN_TRP	--		

Bit	Name	Description	Access	Reset value
[31:10]	--	Reserved	R	0x0
[9]	**STKALIGN**	**Stack Alignment** This bit always reads as 1, indicating 8-byte stack alignment on exception entry. On exception entry, the processor uses bit [9] of the stacked PSR to indicate the stack alignment. On return from the exception, it uses this stacked bit to restore the correct stack alignment.	R	0x1
[8:4]	--	Reserved	R	0x0
[3]	**UNALIGN_TRP**	**Unaligned Access Traps** This bit always reads as 1, indicates that all Unaligned accesses generate a HardFault.	R	0x1
[2:0]	--	Reserved	R	0x0

System Handler Priority Register 2

The SHPR2 register sets the priority level for the SVCall handler.

SHPR2 **System Handler Priority Register 2** **Address : 0xE000 ED1C**

31	30	29	28	27	26	25	24
PRI_11		--					
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--							

Bit	Name	Description	Access	Reset value
[31:30]	**PRI_11**	**Priority Of System Handler 11 – SVCall** “0” denotes the highest priority and “3” denotes the lowest priority.	R/W	0x0
[29:0]	--	Reserved	R	0x0

System Handler Priority Register 3

The SHPR3 register sets the priority level for the SysTick and PendSV handlers.

SHPR3 **System Handler Priority Register 3** **Address : 0xE000 ED20**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
PRI_15		**--**					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14		**--**					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							

Bit	Name	Description	Access	Reset value
[31:30]	**PRI_15**	**Priority Of System Handler 15 – SysTick** “0” denotes the highest priority and “3” denotes the lowest priority.	R/W	0x0
[29:24]	**--**	Reserved	R	0x0
[23:22]	**PRI_14**	**Priority Of System Handler 14 – PendSV** “0” denotes the highest priority and “3” denotes the lowest priority.	R/W	0x0
[29:0]	**--**	Reserved	R	0x0

System Handler Control and State Register

The SHCSR register controls and provides the status of system handlers.

SHCSR **System Handler Control and State Register** **Address : 0xE000 ED24**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SVCALLPENDED	--						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0
[15]	**SVCALLPENDED**	**SVCall Pending bit** This bit reflects the pending state on a read, and updates the pending state, to the value written, on a write. 0 = SVCall is not pending. 1 = SVCall is pending1).	R/W	0x0
[14:0]	--	Reserved	R	0x0

6.3. System timer, SysTick

The processor has a 24-bit system timer, SysTick, that counts down from the reload value to zero, reloads, that is wraps to, the value in the SYST_RVR register on the next clock cycle, then counts down on subsequent clock cycles.

Note: When the processor is halted for debugging, the counter does not decrement.

6.3.1. SysTick usage

The interrupt controller clock updates the SysTick count.

Ensure software uses aligned word accesses to access the SysTick registers. If the SysTick counter reload and current value are undefined at reset, the correct initialization sequence for the SysTick counter is:

1. Program reload value.
2. Clear current value.
3. Program Control and Status register.

Register Map

Base Address : 0xE000_E000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
SYST_CSR	SysTick Control and Status Register	0x010	R/W	0x0000 0000
SYST_RVR	SysTick Reload value Register	0x014	R/W	--
SYST_CVR	SysTick Current value Register	0x018	R/W	--

Registers

SysTick Control and Status Register

SYST_CSR SysTick Control and Status Register Address : 0xE000 E010

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							COUNTFLAG
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--					CLKSOURCE	TICKINT	ENABLE

Bit	Name	Description	Access	Reset value
[31:17]	--	**Reserved**	--	-
[16]	**COUNTFLAG**	Returns 1 if timer counted to 0 since this register is read last time. COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.	R	--
[15:3]	--	**Reserved**	--	-
[2]	**CLKSOURCE**	0 = Clock source is optional (refer to STCLK_S). 1 = Core clock used for SysTick if no external clock provided; this bit will read as 1 and ignore writes.	R/W	0
[1]	**TICKINT**	0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick Current Value register by a register write in software will not cause SysTick to be pended.	R/W	0
[0]	**ENABLE**	0 = The counter is disabled. 1 = The counter will operate in a multi-shot manner.	R/W	0

SysTick Reload Value Register

SYST_RVR		SysTick Reload Value Register					Address : 0xE000 E014
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
RELOAD							
15	14	13	12	11	10	9	8
RELOAD							
7	6	5	4	3	2	1	0
RELOAD							

Bit	Name	Description	Access	Reset value
[31:24]	--	**Reserved**	--	-
[23:0]	**RELOAD**	Value to load into the Current Value register when the counter reaches 0.	R/W	--

SysTick Current Value Register

SYST_CVR SysTick Current Value Register Address : 0xE000 E018

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRENT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURRENT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURRENT							

Bit	Name	Description	Access	Reset value
[31:24]	--	**Reserved**	--	-
[23:0]	CURRENT	Current counter value This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0. Unsupported bits RAZ (see SysTick Reload Value register).	R/W	--

7. NESTED VECTORED INTERRUPT CONTROLLER (NVIC)

7.1. Overview

GPMQ9103A provides an interrupt controller as an integral part of the exception model. The NVIC and the processor core interface are closely coupled, which enables low latency interrupt processing and efficient processing of late arriving interrupts.

7.2. Features

The NVIC supports the following features:

- 32 interrupt nodes
- 4 programmable priority levels for each interrupt node
- Support for interrupt tail-chaining and late-arrival
- Support an external Non Maskable Interrupt (NMI)
- Software interrupt generation

7.3. Exception Types

The exception types are described as below table.

Exception Types	**Descriptions**
Reset	Reset is invoked on power up or a warm reset. The exception model treats reset as a special form of exception. When reset is asserted, the operation of the processor stops, potentially at any point in an instruction. When reset is de-asserted, execution restarts from the address provided by the reset entry in the vector table. Execution restarts in Thread mode.
HardFault	A HardFault is an exception that occurs because of an error during normal or exception processing. HardFaults have a fixed priority of -1, meaning they have higher priority than any exception with configurable priority.
SVCall	A supervisor call (SVC) is an exception that is triggered by the SVC instruction. In an OS environment, applications can use SVC instructions to access OS kernel functions and device drivers.
PendSV	PendSV is an interrupt-driven request for system-level service. In an OS environment, use PendSV for context switching when no other exception is active.
SysTick	A SysTick exception is an exception the system timer generates when it reaches zero. Software can also generate a SysTick exception. In an OS environment, the processor can use this exception as system tick.
Interrupt (IRQ)	A interrupt, or IRQ, is an exception signaled by a peripheral, or generated by a software request. All interrupts are asynchronous to instruction execution. In the system, peripherals use interrupts to communicate with the processor.

Properties of the different exception types

Exception Number	**IRQ Number**	**Exception Type**	**Priority**	**Vector Address**	**Activation**
1	-	Reset	-3	0x0000 0004	Asynchronous
2	-	NMI	-2	0x0000 0008	Asynchronous
3	-13	Hard Fault	-1	0x0000 000C	Synchronous
4 ~ 10	-	Reserved	-	-	-
11	-5	SVCall	Configurable	0x0000 002C	Synchronous
12 ~ 13	-	Reserved	-	-	-
14	-2	PendSV	Configurable	0x0000 0038	Asynchronous
15	-1	SysTick	Configurable	0x0000 003C	Asynchronous
16 and above	0 and above	Interrupt (IRQ)	Configurable	0x0000 0040 and above	Asynchronous

System Interrupt Vector Map

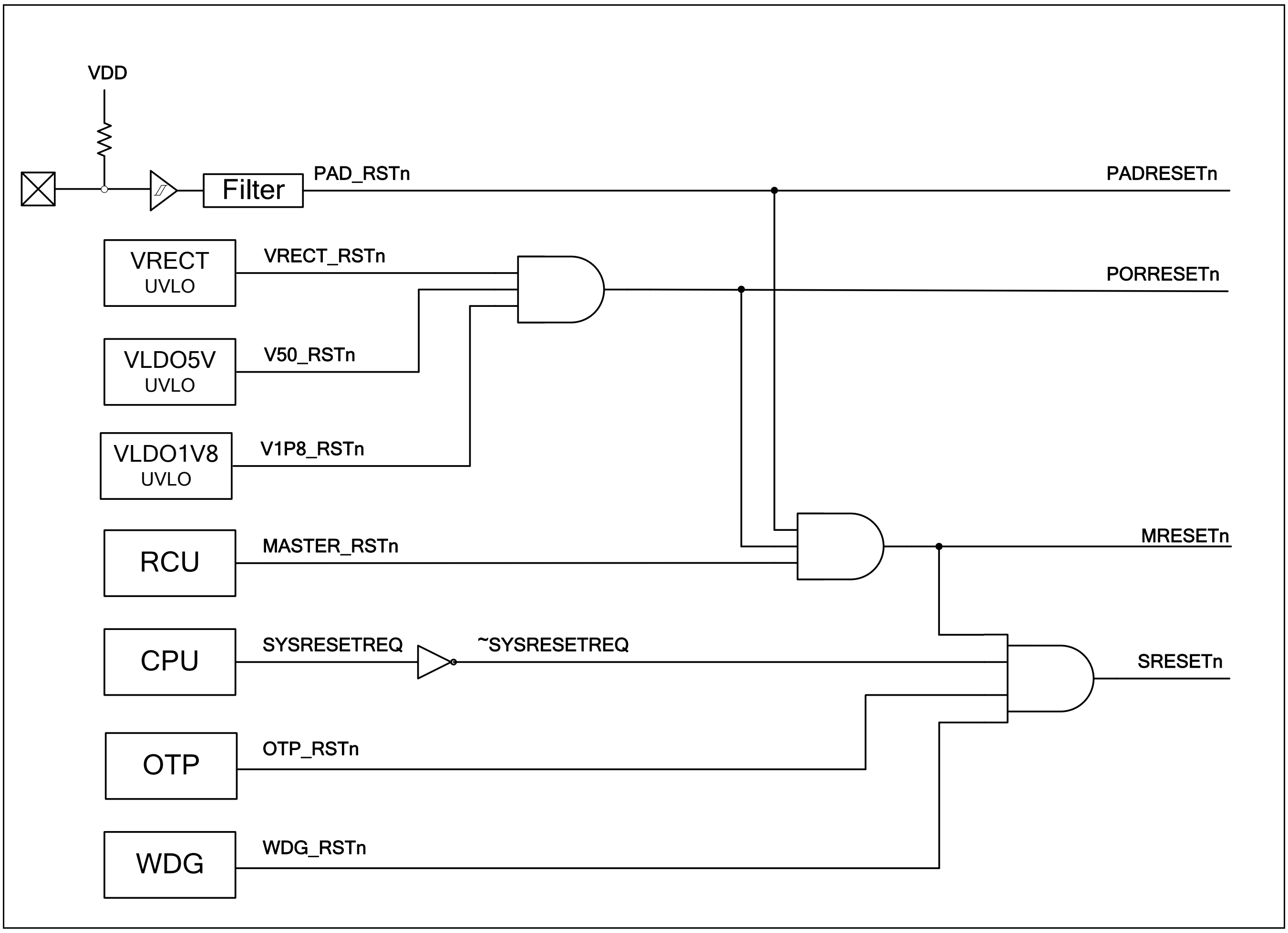
Exception Number	IRQ Number	Interrupt Name	Description	Vector Address
16	0	VRECT_OV_INT	Rectifier Over-Voltage Interrupt	0x0000 0040
17	1	OCL_INT	Over-Current Limit Interrupt	0x0000 0044
18	2	OT_INT	Internal Over-Temperature Interrupt	0x0000 0048
19	3	VOUT_OV_INT	Vout Over Voltage Interrupt	0x0000 004C
20	4	WDT_INT	Watchdog Interrupt	0x0000 0050
21	5	VOUT_CHG_INT	Vout Status Changed Interrupt	0x0000 0054
22	6	ADC0_INT	ADC0 Interrupt	0x0000 0058
23	7	I2C0_INT	I2C0 Interrupt	0x0000 005C
24	8	EXTI0_INT	External Input 0 Interrupt	0x0000 0060
25	9	EXTI1_INT	External Input 1 Interrupt	0x0000 0064
26	10	EXTI2_INT	External Input 2 Interrupt	0x0000 0068
27	11	EXTI3_INT	External Input 3 Interrupt	0x0000 006C
28	12	EXTI4_INT	External Input 4 Interrupt	0x0000 0070
29	13	TMR0_INT	Timer0 Interrupt	0x0000 0074
30	14	TMR1_INT	Timer1 Interrupt	0x0000 0078
31	15	TMR2_INT	Timer2 Interrupt	0x0000 007C
32	16	FSK_TMOUT_INT	AC1/AC2 Timeout interrupt	0x0000 0080
33	17	Reserved		0x0000 0084
34	18	Reserved		0x0000 0088
35	19	Reserved		0x0000 008C
36	20	Reserved		0x0000 0090
37	21	Reserved		0x0000 0094
38	22	Reserved		0x0000 0098
39	23	Reserved		0x0000 009C
40	24	Reserved		0x0000 00A0
41	25	Reserved		0x0000 00A4
42	26	Reserved		0x0000 00A8
43	27	Reserved		0x0000 00AC
44	28	Reserved		0x0000 00B0
45	29	Reserved		0x0000 00B4
46	30	Reserved		0x0000 00B8
47	31	Reserved		0x0000 00BC

8. FUNCTION DESCRIPTIONS

8.1. Reset System

The GPMQ9103A has the following reset types for the system

- PAD Reset, PADRESETn
- Power on Reset, PORRESETn.
- Master Reset, MRESETn
- System Reset, SRESETn



8.1.1. Master Reset

A complete reset to the whole chip is executed by a master reset. Master reset composed of PADRESETn, PORRESETn, MASTER_RSTn and LVR_RSTn.

Low voltage reset, LVR_RSTn, (also known as brown-out reset) is asserted whenever the voltage falls below reset thresholds.

In additional, a MASTER_RSTn can be triggered by setting bit **RCU_CTRL.MRSTn** to reset the whole chip..

8.1.2. System Reset

A system reset affects almost all logics, but exceptions are RCU registers and Debug system if debug link is present.

The debug system is reset by System Reset in normal operation mode when debug link is not present. As debug link present, System Reset would not affect the Debug system.

Register Map

Base Address : 0x5000_3000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
RCU_CTRL	Reset Control Register	0x000	R/W	0x0000 0001
RCU_RSTSTS	Reset Status Register	0x004	R/W	0x0000 0001

Registers

Reset Control Register

Enabling of reset triggered by critical events.

RCU_CTRL **Reset Control Register** **Address : 0x5000 3000**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							**MRSTn**

Bit	Name	Description	Access	Reset value
[31:1]	**--**	**Reserved**	--	-
[0]	**MRSTn**	**Master Reset Trigger (Write Protected)** 0 = Trigger the Master Reset. 1 = No effect. Note: This bit is write protected, refer to SMU.SysUnLock to unlock. It will auto recovery to 1 after program 0 to MRSTn.	R/W	1

Reset Status Register

Reset source triggered flag.

RCU_RSTSTS **Reset Status Register** **Address : 0x5000 3004**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--	WDGRF	--		SYSRF	MRF	PADRF	PORRF

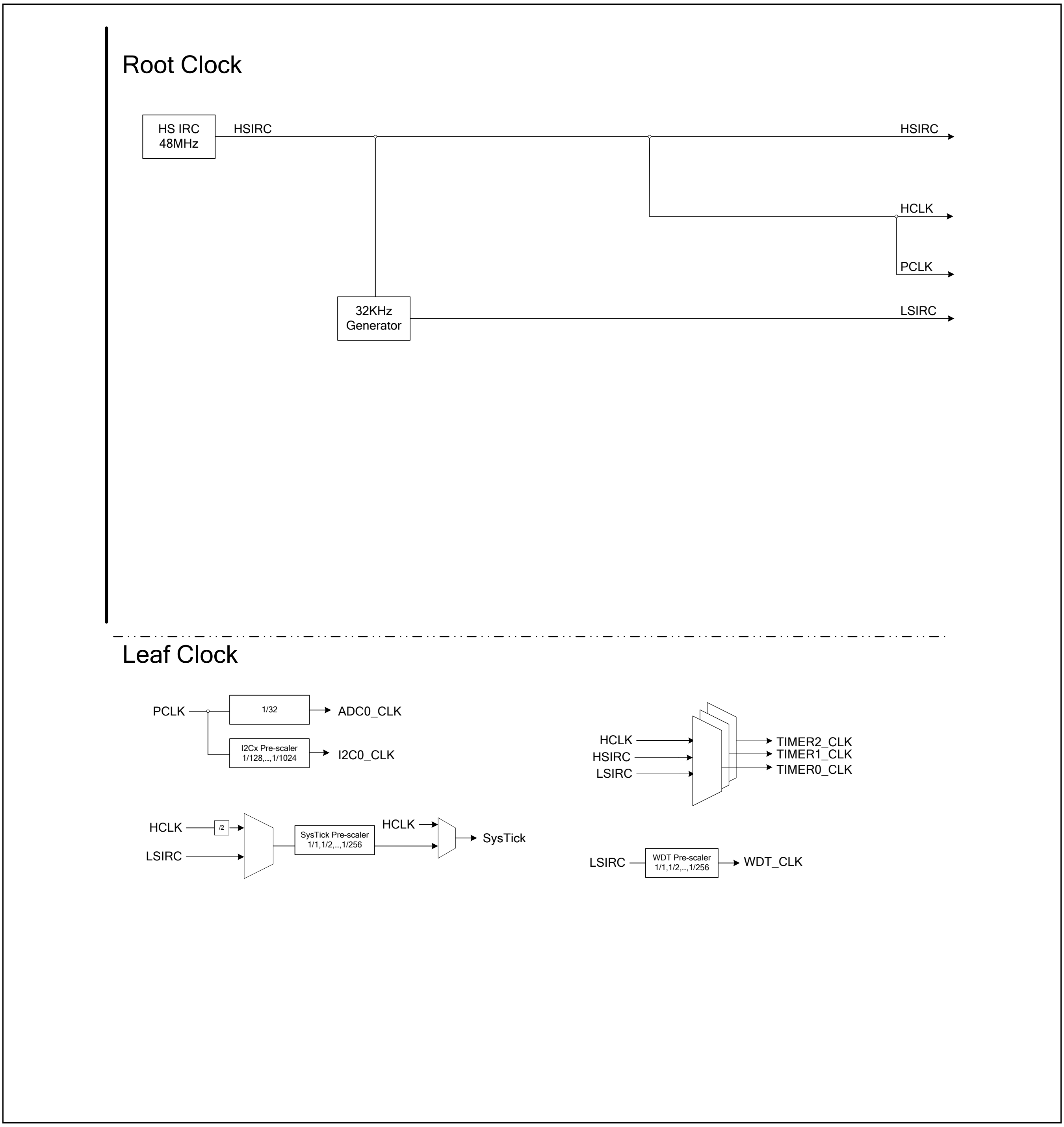
Bit	Name	Description	Access	Reset value
[31:7]	--	**Reserved**	--	0
[6]	**WDGRF**	**Watchdog Reset Flag** 0 = No reset from watchdog reset. 1 = Watchdog reset occurs. Note: Write 1 to clear this bit	R/W	0
[5:4]	--	**Reserved**	--	0
[3]	**SYSRF**	**SYSRESETREQ Reset Flag** 0 = No reset from SYSRESETREQ reset. 1 = SYSRESETREQ reset occurs. Note: Write 1 to clear this bit	R/W	0
[2]	**MRF**	**Software Triggered Master Reset Flag** 0 = No reset from software triggered master reset. 1 = Software triggered reset occurs. Note: Write 1 to clear this bit.	R/W	0
[1]	**PADRF**	**External PAD Reset Flag** 0 = No reset from external reset pad. 1 = External PAD reset occurs. Note: Write 1 to clear this bit.	R/W	0
[0]	**PORRF**	**POR Reset Flag** 0 = No reset from Power-On Reset. 1 = Power-On Reset occurs. Note: Write 1 to clear this bit.	R/W	1

8.2. Clock Control Unit

The clock control unit has two clock sources:

- Low Speed Internal RC Oscillator 32KHz
- High Speed Internal RC Oscillator 48MHz

8.2.1. Clock Tree Diagram



Register Map

Base Address : 0x5000_1000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
CLK_AHBCKEN	AHB Peripherals Clock Enable Register	0x000	R/W	0x0000 0000
CLK_APBCKEN	APB Peripherals Clock Enable Register	0x010	R/W	0x0020 0000
CLK_AHBCKSEL0	AHB Peripherals Clock Selection 0 Register	0x020	R/W	0x0000 0000
CLK_CLKDIV	Clock Divider Register	0x040	R/W	0x0000 0000
CLK_CLKSTS	Clock Status Register	0x050	R	0x0000 0000

Registers

AHB Peripherals Clock Enable Register

CLK_AHBCKEN **AHB Peripherals Clock Enable Register** **Address : 0x5000 1000**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--					TMR2CKEN	TMR1CKEN	TMR0CKEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--						SYSTICKEN	--

Bit	Name	Description	Access	Reset value
[31:19]	--	**Reserved**	R	0
[18]	**TMR2CKEN**	**Timer 2 Clock Enable** 0 = Disabled 1 = Enabled	R/W	0
[17]	**TMR1CKEN**	**Timer 1 Clock Enable** 0 = Disabled 1 = Enabled	R/W	0
[16]	**TMR0CKEN**	**Timer 0 Clock Enable** 0 = Disabled 1 = Enabled	R/W	0
[15:2]	--	**Reserved**	R	0
[1]	**SYSTICKEN**	**SysTick External Clock Enable** 0 = Disabled 1 = Enabled	R/W	0
[0]	--	**Reserved**	R	0

APB Peripherals Clock Enable Register

CLK_APBCKEN APB Peripherals Clock Enable Register Address : 0x5000 1010

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--			CKOCKEN	--			
15	14	13	12	11	10	9	8
--			I2C0CKEN	--			
7	6	5	4	3	2	1	0
--							ADC0CKEN

Bit	Name	Description	Access	Reset value
[31:21]	--	**Reserved**	R	0
[20]	CKOCKEN	**Clock Output Enable** 0 = Disabled 1 = Enabled		
[19:13]	--	**Reserved**	R	0
[12]	I2C0CKEN	**I2C Controller 0 Clock Enable** 0 = Disabled 1 = Enabled	R/W	0
[11:1]	--	**Reserved**	R	0
[0]	ADC0CKEN	**ADC Controller 0 Clock Enable** 0 = Disabled 1 = Enabled	R/W	0

AHB Peripherals Clock Selection 0 Register

CLK_AHBCKSEL0	AHB Peripherals Clock Selection 0 Register						Address : 0x5000 1020
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--		STCKSEL		--			

Bit	Name	Description	Access	Reset value
[31:6]	--	**Reserved**	R	0
[5:4]	**STCKSEL**	**SysTick External Clock Source Selection** 00 = HCLK/2. 01 = Reserved. 10 = LSIRC (Low Speed Internal RC) 11 = Reserved.	R/W	0
[3:0]	--	**Reserved**	R	0

Clock Divider Register

CLK_CLKDIV Clock Divider Register Address : 0x5000 1040

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
STICKDIV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R	0
[15:8]	**STICKDIV**	**External SysTick Clock Divider** External SysTick clock = (Mux Output Clock) / (STICKDIV + 1)	R/W	0
[7:0]	--	**Reserved**	R	0

Clock Status Register

CLK_CLKSTS **Clock Status Register** **Address : 0x5000 1050**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--						LSIRCRDY	HSIRCRDY

Bit	Name	Description	Access	Reset value
[31:2]	--	**Reserved**	R	0
[1]	**LSIRCRDY**	**Low Speed Internal RC Ready Flag** 0 = LSIRC is not ready. 1 = LSIRC is ready.	R	0
[0]	**HSIRCRDY**	**High Speed Internal RC Ready Flag** 0 = HSIRC is not ready. 1 = HSIRC is ready.	R	0

8.3. System Management Unit

The system management unit included user defined ID information, instruction fetch accelerator and some control setting.

Register Map

Base Address : 0x5000_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
SMU_SYSLOCK	System Control Signal Lock Register	0x000	R/W	0
SMU_PID	Produce ID Register	0x004	R	--

Registers

System Control Signal Lock Register

SMU_SYSLOCK System Control Signal Lock Register Address : 0x5000 0000

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							SysUnLOCK
7	6	5	4	3	2	1	0
UnLOCK_KEY							

Bit	Name	Description	Access	Reset value
[31:9]	--	**Reserved**	R	0
[8]	**SysUnLOCK**	**UnLOCK Status** 0 = Lock. 1 = UnLock. Note: This bit status is based on UnLOCK_KEY operation.	R	0
[7:0]	**UnLOCK_KEY**	**Specified System Control UnLock Key** Write 0xAB and 0x12 to this UnLOCK_KEY consequency, the specified registers will be unlock, Note: If **SysUnLOCK** in unlocked state, write other words except 0x12 to UnLOCK_KEY, the specified registers will be locked. Specified Lock Protect Register as below **ACU_REG_CTRL** **ACU_RECT_CTRL**	R/W	0

Product ID Register

SMU_PID Product ID Register Address : 0x5000 0004

31	30	29	28	27	26	25	24
PID							
23	22	21	20	19	18	17	16
PID							
15	14	13	12	11	10	9	8
PID							
7	6	5	4	3	2	1	0
PID							

Bit	Name	Description	Access	Reset value
[31:0]	**PID**	**Product ID of Chip**	R	--

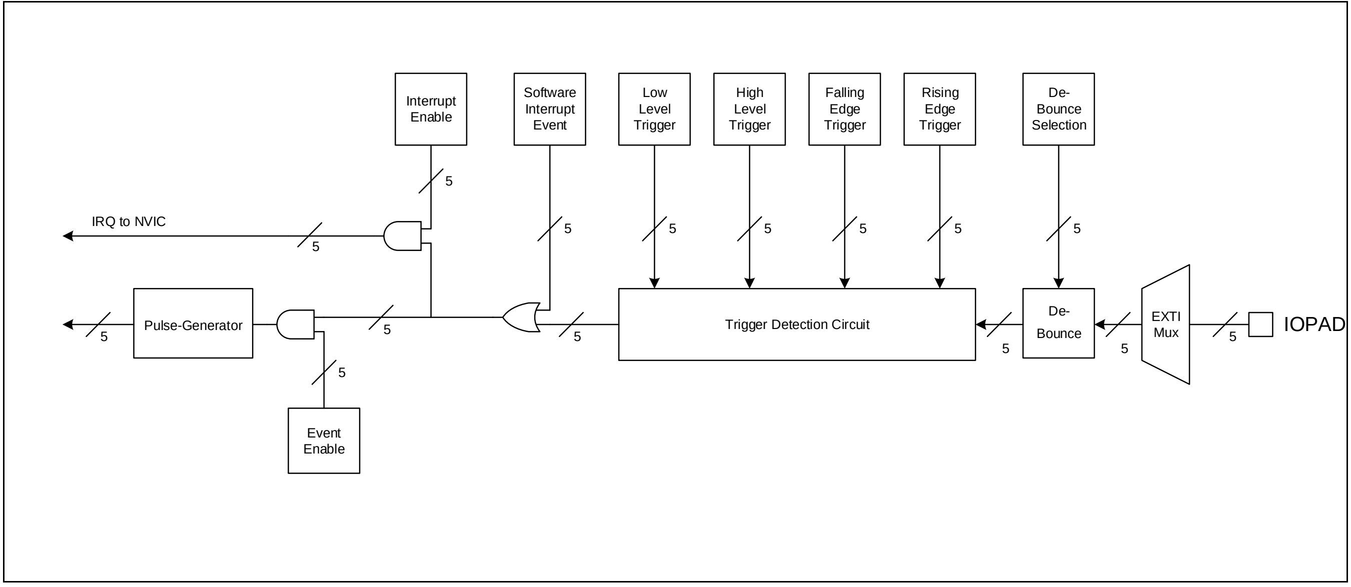
8.4. Interrupt Control Unit

Interrupt control unit include NMI interrupt source configuration, software interrupt request setting and external interrupt/event control. The external interrupt/event controller consists of up to 5 edge detectors in connectivity line devices. Each input line can be independently configured to select the trigger event (rising or falling or both or level) and generated event pulse when event control enabled.

8.4.1. NMI interrupts source selection

NMI interrupt source of GPMQ9103A can be configured as one of NVIC IRQ Numbers. The NMISRC are corresponded to NVIC IRQ Numbers.

8.4.2. External interrupt/event controller block diagram



Functional description

To generate the interrupt, the interrupt line should be configured and enabled. This is done by programming the trigger registers with the desired edge detection or level detection and by enabling the interrupt request by writing a ‘1’ to the corresponding bit in the interrupt enable register. When the selected detection condition occurs on the external interrupt line, an interrupt request is generated.

To generate the event, the event line should be configured and enabled. This is done by programming the trigger registers with the desired edge detection or level detection and by enabling the event request by writing a ‘1’ to the corresponding bit in the event enable register. When the selected detection condition occurs on the event line, an event pulse is generated.

An interrupt/event request can also be generated by software by writing a ‘1’ in the software interrupt/event register.

Register Map

Base Address : 0x5000_4000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
ITU_NMICTRL	NMI Interrupt Control Register	0x000	R/W	0x0000 0000
ITU_IRQ	Interrupt Request Register	0x004	R/W	0x0000 0000
ITU_EXTIEN	External Interrupt Enable Register	0x010	R/W	0x0000 0000
ITU_EXTEEN	External Event Enable Register	0x014	R/W	0x0000 0000
ITU_EXTRHT	External Rising Edge / High Level Trigger Register	0x018	R/W	0x0000 0000
ITU_EXTFLT	External Falling Edge / Low Level Trigger Register	0x020	R/W	0x0000 0000
ITU_EXTIFLG	External Interrupt Flag Register	0x028	R/W1C	0x0000 0000
ITU_EXTEFLG	External Event Flag Register	0x02C	R/W1C	0x0000 0000
ITU_EXTSWIE	External Software Trigger Interrupt Event Register	0x030	R/W1C	0x0000 0000
ITU_EXTDEB	External De-bounce Enable Register	0x034	R/W	0x0000 0000

Registers

NMI Interrupt Control Register

ITU_NMICTRL **NMI Interrupt Control Register** **Address : 0x5000 4000**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							NMIEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
				NMISRC			

Bit	Name	Description	Access	Reset value
[31:17]	--	**Reserved**	R	0
[16]	**NMIEN**	**NMI Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0
[15:4]	--	**Reserved**	R	0
[3:0]	**NMISRC**	**NMI Source Selection** NMI Source numbers are referred to IRQ number of System Interrupt Vector Map.	R/W	0

Interrupt Request Register

ITU_IRQ **Interrupt Request Register** **Address : 0x5000 4004**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IRQ							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IRQ							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R	0
[15:0]	**IRQ**	**Interrupt Request** When the IRQ[n] is 0, setting IRQ[n] to 1 will generate an interrupt to Cortex™-M0 NVIC[n] When the IRQ[n] is 1 (mean an interrupt is assert), setting 1 to the MCU_bit[n] will clear the interrupt and setting IRQ[n] 0 has no effect Note: Refer to **ICSR.** VECTACTIVE to check the currently active exception.	R/W	0

External Interrupt Enable Register

ITU_EXTIEN **External Interrupt Enable Register** **Address : 0x5000 4010**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			EXTIEN				

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4:0]	**EXTIEN**	**External Input Interrupt Enable** Bit number of EXTIEN means corresponding port. 0 = External Interrupt request is disabled. 1 = External Interrupt request is enabled.	R/W	0

External Input Event Mask Register

ITU_EXTEEN **External Input Event Enable Register** **Address : 0x5000 4014**

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--			EXTEEN				

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4:0]	**EXTEEN**	**External Input Event Enable** Bit number of EXTEEN means corresponding port. 0 = External event request is disabled. 1 = External event request is enabled.	R/W	0

External Input Rising Edge / High Level Trigger Register

ITU_EXTRHT External Input Rising Edge / High Level Trigger Register Address : 0x5000 4018

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--			EXTHT				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			EXTRT				

Bit	Name	Description	Access	Reset value
[31:21]	--	**Reserved**	R	0
[20:16]	**EXTHT**	**External Input High Level Trigger Enable** Bit number of EXTHT means corresponding port. 0 = External high level trigger is disabled. 1 = External high level trigger is enabled.	R/W	0
[15:5]	--	**Reserved**	R	0
[4:0]	**EXTRT**	**External Input Rising Trigger Enable** Bit number of EXTRT means corresponding port. 0 = External rising trigger is disabled. 1 = External rising trigger is enabled.	R/W	0

External Input Falling Edge / Low Level Trigger Register

ITU_EXTFLT External Input Falling Edge / Low Level Trigger Register Address : 0x5000 4020

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--			**EXTLT**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			**EXTFT**				

Bit	Name	Description	Access	Reset value
[31:21]	**--**	**Reserved**	R	0
[20:16]	**EXTLT**	**External Input Low Level Trigger Enable** Bit number of EXTLT means corresponding port. 0 = External low level trigger is disabled. 1 = External low level trigger is enabled.	R/W	0
[15:5]	**--**	**Reserved**	R	0
[4:0]	**EXTFT**	**External Input Falling Trigger Enable** Bit number of EXTFT means corresponding port. 0 = External falling trigger is disabled. 1 = External falling trigger is enabled.	R/W	0

External Input Interrupt Flag Register

ITU_EXTIFLG External Input Interrupt Flag Register Address : 0x5000 4028

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			EXTIFLG				

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4:0]	**EXTIFLG**	**External Interrupt Flag** Bit number of EXTIFLG means corresponding port. 0 = No trigger request occurred. 1 = Select request occurred. Note : Cleared by writing 1..	R/W1C	0

External Event Flag Register

ITU_EXTEFLG External Event Flag Register Address : 0x5000 402C

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			EXTEFLG				

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4:0]	**EXTEFLG**	**External Event Flag** Bit number of EXTEFLG means corresponding port. 0 = No trigger request occurred. 1 = Select request occurred. Note: Cleared by writing 1.	R/W1C	0

External Software Interrupt Event Trigger Register

ITU_EXTSWIE **External Software Interrupt Event Trigger Register** **Address : 0x5000 4030**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			EXTSWIE				

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4:0]	**EXTSWIE**	**External Software Interrupt Event Trigger** Bit number of EXTSWIE means corresponding port. 0 = No software trigger request occurred. 1 = Selected software request occurred. Note: Cleared by writing 1.	R/W1C	0

External De-bounce Enable Register

ITU_EXTDEB **External De-bounce Enable Register** **Address : 0x5000 4034**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--					DEBSEL		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			DEBEN				

Bit	Name	Description	Access	Reset value
[31:19]	--	**Reserved**	R	0
[18:16]	**DEBSEL**	**External De-bounce Enable Flag** 000 = De-bounce window is 2 cycles of HSIRC 8MHz 001 = De-bounce window is 4 cycles of HSIRC 8MHz 010 = De-bounce window is 8 cycles of HSIRC 8MHz 011 = De-bounce window is 16 cycles of HSIRC 8MHz 100 = De-bounce window is 32 cycles of HSIRC 8MHz 101 = De-bounce window is 64 cycles of HSIRC 8MHz 110 = De-bounce window is 128 cycles of HSIRC 8MHz 111 = De-bounce window is 256 cycles of HSIRC 8MHz	R/W	0
[4:0]	**DEBEN**	**External De-bounce Enable Flag** Bit number of EXTDEB means corresponding port. 0 = External input de-bounce disabled. 1 = External input de-bounce enabled.	R/W	0

8.5. Analog Control Unit

8.5.1. Overview

Analog control unit is designed for some analog devices such as below description.

8.5.2. Internal RC

The HSIRC8M clock signal is generated from an internal 48 MHz RC Oscillator and can be used directly as a system clock.

The HSIRCRDY flag in the **CLK_CLKSTS .HSIRCRDY** indicates if the high-speed external oscillator is stable or not.

Register Map

Base Address : 0x5000_5000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
ACU_REG_CTRL	Regulator Control Register	0x000	R/W	0x0000 0000
ACU_RECT_CTRL	Synchronous Rectifier Control Register	0x010	R/W	0x0000 0000

Registers

Regulator Control Register

ACU_REG_CTRL **Regulator Control Register** **Address : 0x5000 5000**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--					VOUT_STS	VOUT_OCL	VOUT_OV
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VOUT_MODE	LDOVOUT_CL		LDOVOUT_TURN		LDOVOUT_SEL		LDOVOUT_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			LDO5V_OK	--		LDO5V_SEL	

Bit	Name	Description	Access	Reset value
[31:19]	--	**Reserved**	R	0
[18]	**VOUT_STS**	**VOUT Status.** 0 = VOUT OFF. 1 = VOUT ON.	R	0
[17]	**VOUT_OCL**	**VOUT Over Current Limit.** 0 = VOUT Source Current < LDOVOUT_CL 1 = VOUT Source Current > LDOVOUT_CL	R	0
[16]	**VOUT_OV**	**VOUT Over Voltage.** 0 = VOUT Voltage < (LDOVOUT_SEL * 1.25) 1 = VOUT Voltage > (LDOVOUT_SEL * 1.25)	R	0
[15]	**VOUT_MODE**	**VOUT Mode Selection. (Write Protect)** 0 = LDO Mode 1 = Switch Mode	R/W	0
[14:13]	**LDOVOUT_CL**	**LDOVOUT Current Limit Selection. (Write Protect)** 00 = 1A 01 = 1.5A 10 = 2A 11 = 0.5A	R/W	0
[12:11]	**LDOVOUT_TURN**	**LDOVOUT Output Voltage Fine Turn. (Write Protect)** 00 = 0% 01 = +2% 10 = +4% 11 = -2%	R/W	0
[10:9]	**LDOVOUT_SEL**	**LDOVOUT Output Voltage Selection. (Write Protect)** 00 = 5V 01 = 7V 10 = 9V 11 = 12V	R/W	0
[8]	**LDOVOUT_EN**	**LDOVOUT Enable (Write Protect)** 0 = Shout down 1 = Enable	R/W	0
[7:5]	--	**Reserved**	R	0
[4]	**LDO5V_OK**	**LDO5V Voltage Status** 0 = LDO5V voltage < LDO5V_UVLO 1 = LDO5V voltage > LDO5V _UVLO	R	0
[3:2]	--	**Reserved**	R	0
[1:0]	**LDO5V_SEL**	**LDO5V Output Voltage Selection (Write Protect)** 00 = 5.2V 01 = 4.5V 10 = 5.5V 11 = 5.0v	R/W	0

Rectifier Control Register

ACU_RECT_CTRL **Rectifier Control Register** **Address : 0x5000 5010**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--			**VRECT_OK**	**AC_ERR**	**RECT_OVP**	**HS_STS**	**LS_STS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--		**RECT_OVP**		**RECT_MODE**		**VRECT_ITH**	

Bit	Name	Description	Access	Reset value
[31:13]	**--**	**Reserved**	R	0
[12]	**VRECT_OK**	**Rectifier Voltage Status** 0 = Rectifier voltage < RECT_UVLO 1 = Rectifier voltage > RECT_UVLO	R	0
[11]	**AC_ERR**	**Rectifier Error Status** 0 = Normal 1 = Rectifier Error Occurs Note: If AC_ERR =1, do not turn on VOUT, and ask Tx to ping again.	R	0
[10]	**RECT_OVP**	**Rectifier Over Voltage Status** 0 = Rectifier voltage < RECT_OVP 1 = Rectifier voltage > RECT_OVP	R	0
[9]	**HS_STS**	**Rectifier high -side switch activation status** 0 = Off 1 = Activated.	R	0
[8]	**LS_STS**	**Rectifier low-side switch activation status** 0 = Off 1 = Activated.	R	0
[7:6]	**--**	**Reserved**	R	0
[5:4]	**RECT_OVP**	**Rectifier Over Voltage Protection Level Selection (Write Protect)** 00 = 15V 01 = 17V 10 = 19V 11 = 21V	R/W	0
[3:2]	**RECT_MODE**	**Rectifier Td, Vt mode selection. (Write Protect)** 00 = Trimmed Vt ; Td_ON=11, Td_OFF=00 01 = Trimmed Vt ; Trimmed Td 10 = Vt prone to be over ; Td_ON=11, Td_OFF=00 11 = Vt prone to be over ; Trimmed Td	R/W	0
[1:0]	**RECT_ITH**	**Rectifier high-side switch activated when IOUT > ITH. (Write Protect)** 00 = 200mA 01 = 250mA 10 = 300mA 11 = 400mA	R/W	0

8.6. Non-Volatile Memory Controller

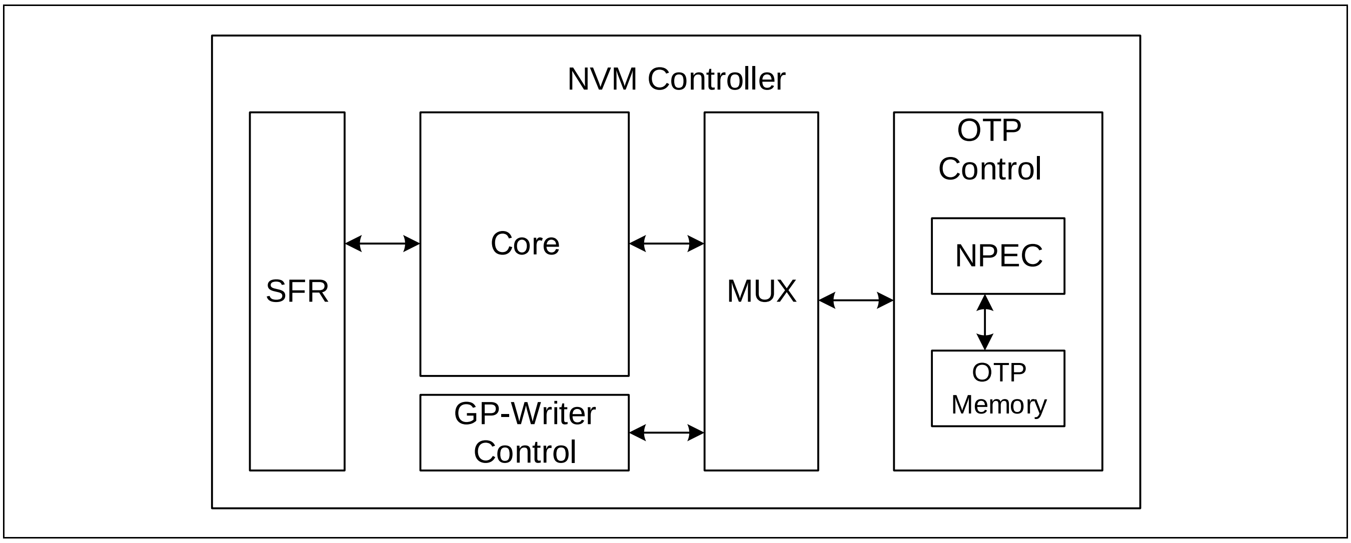
8.6.1. Overview

The GPMQ9103A has an embedded user-programmable non-volatile memory (NVM), also called OTP, for storage of user code and data. NVM of GPMQ9103A composed of Program Memory. Program Memory where is user application code stored.

8.6.2. Features

- Reading by word (4 Bytes)
- Erasing by page (512 Bytes)
- Embedded 16KB Program Memory

8.6.3. Block Diagram

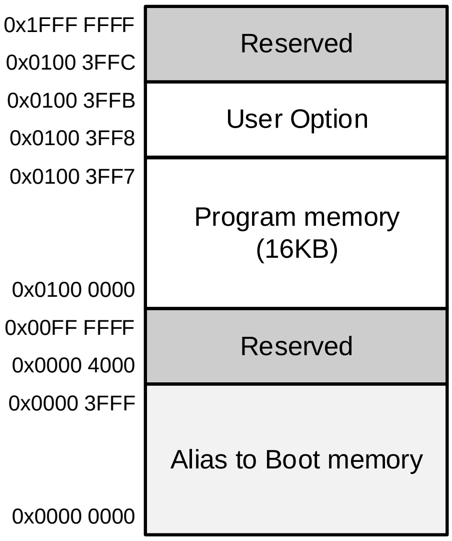


8.6.4. Function Description

Non-volatile memory controller includes the memory organization, data memory, Boot mode selection, NVM program controller (NPEC) and User options.

8.6.5. Memory Organization

Memory organization consists of the Program memory, Boot memory and User option. After booting, the OTP memory will be alias to address 0x0 based on the boot mode selection.



8.6.6. User Option Memory

User Option Memory is user programmable configuration area for OTP security lock. It is loaded from user option memory to its corresponding control registers during chip power on.

8.6.7. NVM Program Controller

The NVM Program Controller (NPEC) block handles the program operations of the embedded user-programmable non-volatile memory (NVM).

An ongoing NVM operation will not block the CPU as long as the CPU does not access the OTP memory.

8.6.7.1. Key values

The key values are as follows:

- RDPRT key != **0x0**
- KEY1 = **0xABCD5678**
- KEY2 = **0x1234FEDC**

RDPRT key is used for USER_SLOCK register to block the OTP memory content be read out.

KEY1 and KEY2 are used to unlock some specified bits of NVM_CTRL register.

8.6.7.2. Unlocking the NPEC

After reset, the NPEC block is protected. The NVM_CTRL register is not accessible in write mode. An unlocking sequence should be written to the NVM_FSHKEY register to open up the NPEC block. This sequence consists of two write cycles, where two key values (KEY1 and KEY2) are written to the NVM_FSHKEY address.

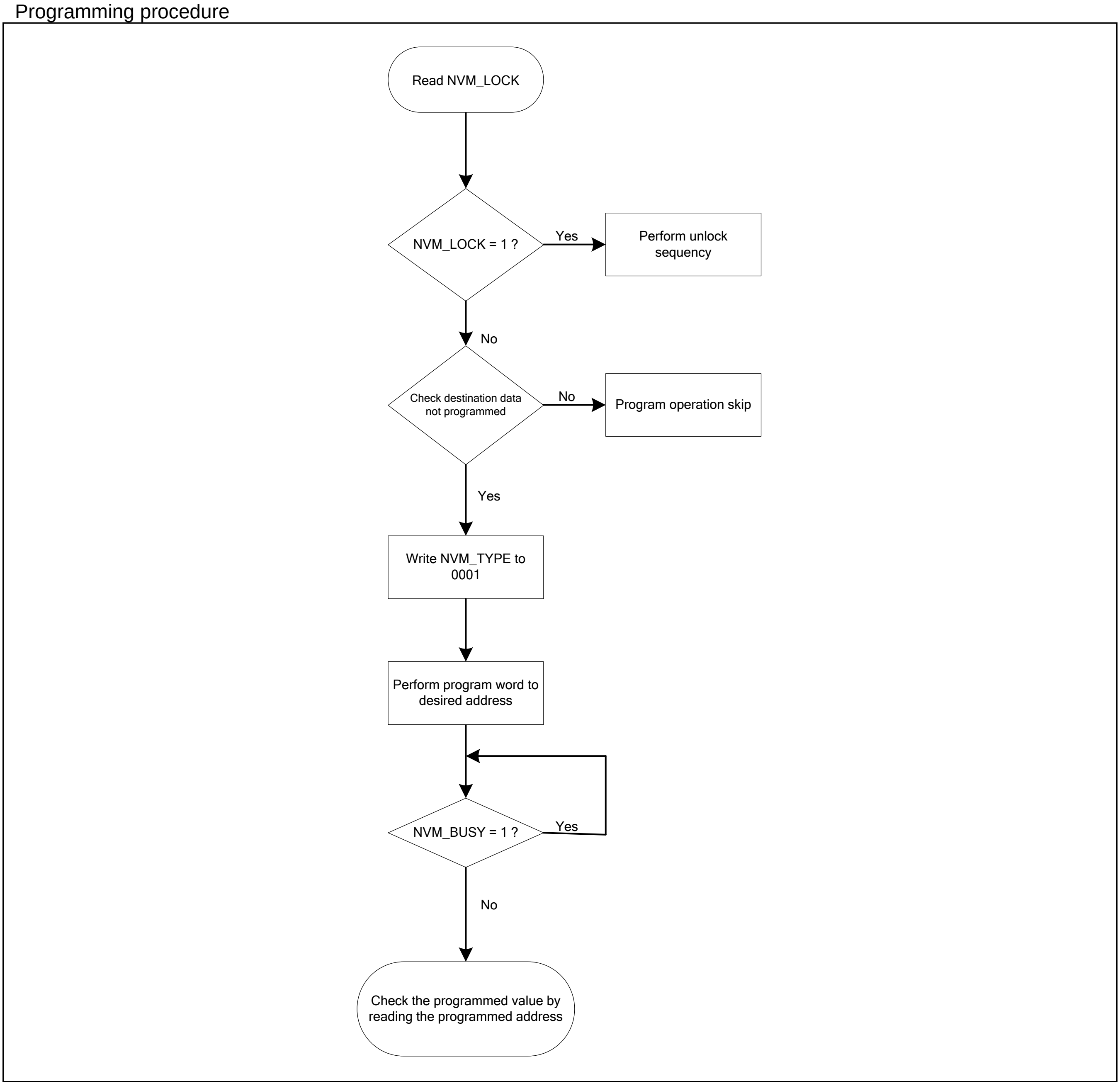
Any wrong sequence locks up the NPEC block and NVM_CTRL register until the next reset. The NPEC block and NVM_CTRL register can be locked by the user’s software by writing then NVM_LOCK bit of the NVM_CTRL register to 1. In this case, the NPEC can be unlocked by writing the correct sequence of keys into NVM_FSHKEY.

8.6.7.3. Non Volatile Memory Programming

The program operation is started when the CPU writes a word into a program memory address with the NVM_TYPE of the NVM_CTRL register set in Program. During the programming, NVM_BUSY bit set, the CPU will keep wait until the ongoing OTP memory programming is completed.

To program memory, the procedure as below should be followed:

Programming procedure


Read NVM_LOCK
NVM_LOCK = 1 ?
Yes
Perform unlock sequency
No
Check destination data not programmed
No
Program operation skip
Yes
Write NVM_TYPE to 0001
Perform program word to desired address
NVM_BUSY = 1 ?
Yes
No
Check the programmed value by reading the programmed address

Registers Map

Base Address : 0x0F00_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
USER_OPT0	User Option 0 Control Register	0x000	W	--

Base Address : 0x5001_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
NVM_FSHKEY	OTP Update Key Register	0x000	R/W	0x0000 0000
NVM_CTRL	NVM Operation Control Register	0x008	R/W	0x0000 0001
NVM_STS	NVM Operation Status Register	0x010	R	0x0000 0000
NVM_UOPT0STS	User Option 0 Status	0x014	R	--

Registers

User Option0 Control Register

USER_OPT0 **User Option0 Control Register** **Address : 0x0100 3FF8**

31	30	29	28	27	26	25	24
USER_SLOCK							
23	22	21	20	19	18	17	16
USER_SLOCK							
15	14	13	12	11	10	9	8
USER_SLOCK							
7	6	5	4	3	2	1	0
USER_SLOCK							

Bit	Name	Description	Access	Reset value
[31:0]	**USER_SLOCK**	**Security Lock Selection** 0x0 = OTP memory can be read out by ICE or writer. Others = OTP memory can't be read or written by ICE or writer.	R/W	--

OTP Update Key Register

NVM_FSHKEY **OTP Update Key Register** **Address : 0x5001 0000**

31	30	29	28	27	26	25	24
NVM_FSHKEY							
23	22	21	20	19	18	17	16
NVM_FSHKEY							
15	14	13	12	11	10	9	8
NVM_FSHKEY							
7	6	5	4	3	2	1	0
NVM_FSHKEY							

Bit	Name	Description	Access	Reset value
[31:0]	**NVM_FSHKEY**	**OTP Control Register Unlock Key** Write 0xABCD5678 and 0x1234FEDC to NVM_FSHKEY in sequence, the NVM_CTRL registers will be unlock, all bit fields can be accessed. Note: When write other words to NVM_FSHKEY, the NVM_CTRL register will be reset and locked.	R/W	0x0000 0000

NVM Operation Control Register

NVM_CTRL **NVM Operation Control Register** **Address : 0x5001 0008**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--				NVM_TYPE			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							NVM_LOCK

Bit	Name	Description	Access	Reset value
[31:12]	--	**Reserved**	--	0
[11:8]	**NVM_TYPE**	**NVM Operation Type** 0000 = Idle or Read. 0001 = Program Others = Reserved Note : NVM_TYPE writes access is available on writing the correct key sequence to the NVM_FSHKEY. Note : NVM Program Type only support 32-bit program, other data size will not affect occurs.	R/W	0
[7:1]	--	**Reserved**	--	0
[0]	**NVM_LOCK**	**NVM_CTRL and NPEC status** Write to 1 only. 0 = Indicate the NVM_CTRL are unlocked. 1 = Indicate the NVM_CTRL are locked. Note : This bit is reset by hardware after detecting the unlock sequence of NVM_FSKEY.	R/W	1

NVM Operation Status Register

NVM_ADDR		NVM Operation Status Register					Address : 0x5001 0010
31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0
							NVM_BUSY

Bit	Name	Description	Access	Reset value
[31:1]	--	**Reserved**	--	0
[0]	NVM_BUSY	**NVM Operation Busy Flag** This indicates that a NVM operation is in progress. This is set on the beginning of a NVM operation and reset when the operation finishes or when an error occurs.	R	0

NVM User Operation 0 Status

NVM_UOPT0STS NVM User Operation 0 Status Address : 0x5001 0014

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--		**BOOT_STS**		**--**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							**SLOCK_STS**

Bit	Name	Description	Access	Reset value
[31:9]	**--**	**Reserved**	--	0
[13:12]	**BOOT_STS**	**Boot Mode Status** 1x = Boot from OTP Memory. Note : This field is reflect from **USER_OPT0.USER_BOOT** in Lock or unlock state.	R	--
[11:1]	**--**	**Reserved**	--	0
[0]	**SLOCK_STS**	**Security Lock Status** 0 = Security UnLock. 1 = Security Lock. Note : This field is reflect from **USER_OPT0.USER_SLOCK** in Lock or unlock state.	R	--

8.7. Timer

8.7.1. Overview

The GPMQ9103A incorporates a counter module, called TMU. The TMU unit is comprised of three 16 bit Timer slices, TMx (x=0,1,2). Each Timer Slice can work in compare or capture mode, TM2 don’t support capture mode. In compare mode, one slice has a dedicated compare channels. In capture mode, one capture registers is available. Each TMx module supports one service request lines that can be triggered ADC conversion.

8.7.2. Features

Each TMU represents a combination of three timer slices that can work independently in compare or capture mode. The functions as below.

General Features

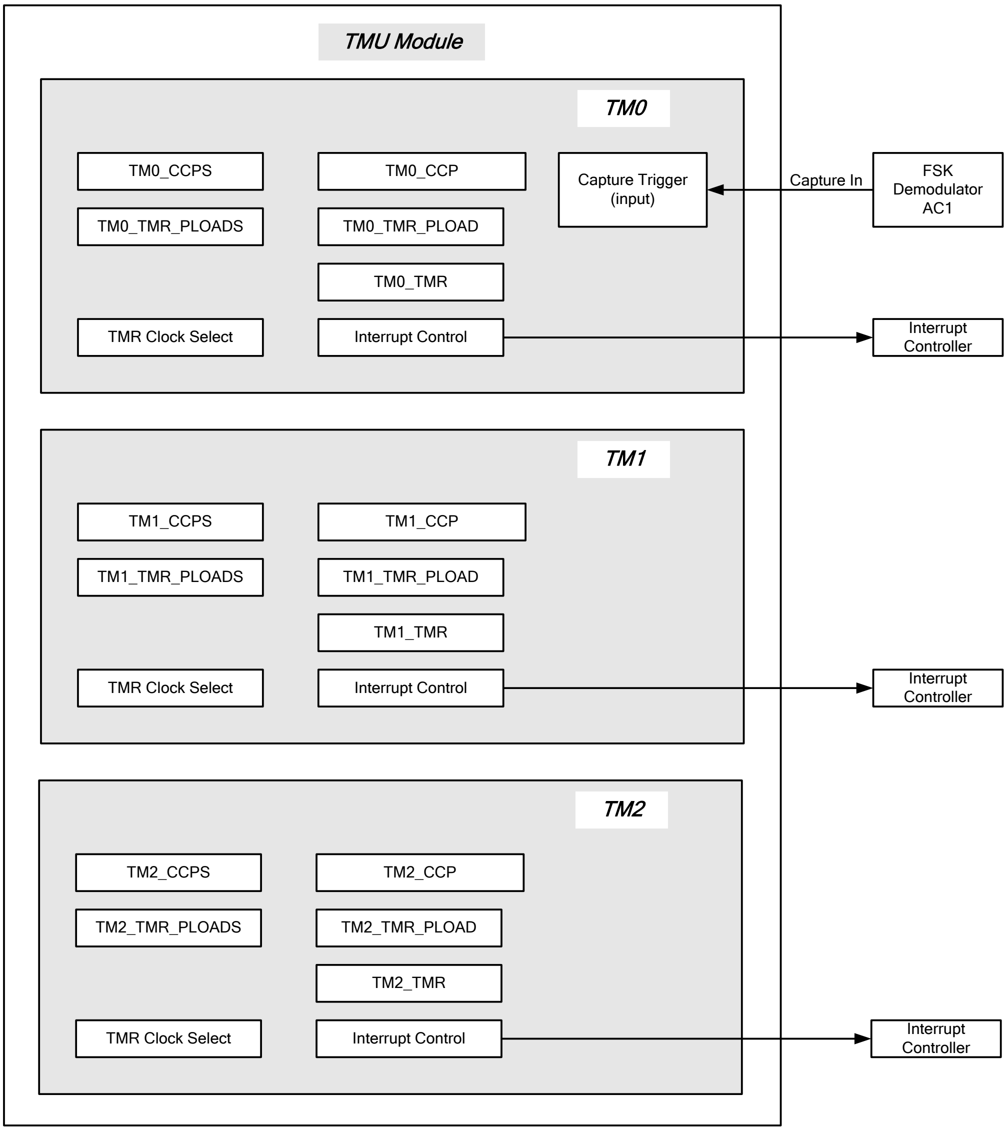
- 16-bit up, down, auto-reload counter
- 14 counter clock selected
- Shadow transfer for the period, compare channels
- An independent channels for:
  - Input Capture
  - Output Compare
- Programmable low pass filter for the capture mode inputs
- Normal timer mode

Additional features

- Interrupt generation on the following events:
  - Update: counter overflow/underflow
  - Input capture
  - Output compare

8.7.3. Block Diagram

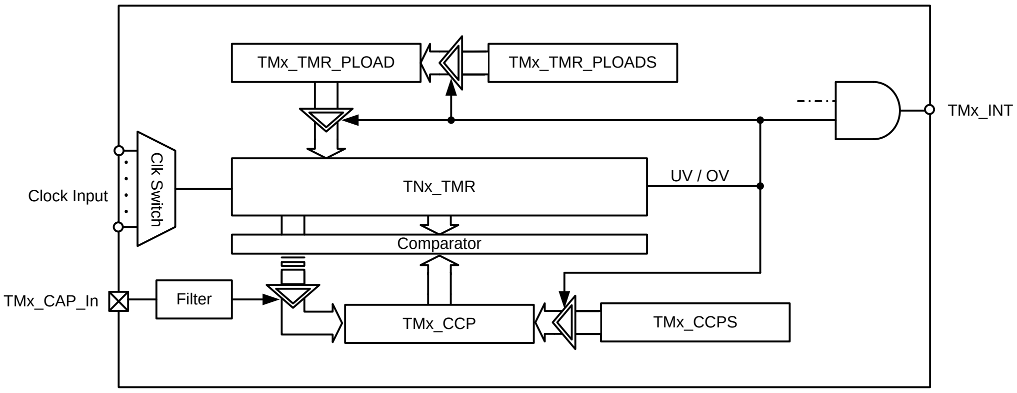
Each TMx timer slice can operate independently for all the available modes.



TMU block diagram

8.7.4. Function Description

The input path of a TMU slice is comprised of a GPIO alternate. The output path contains an interrupt trigger event.



TMx slice block diagram

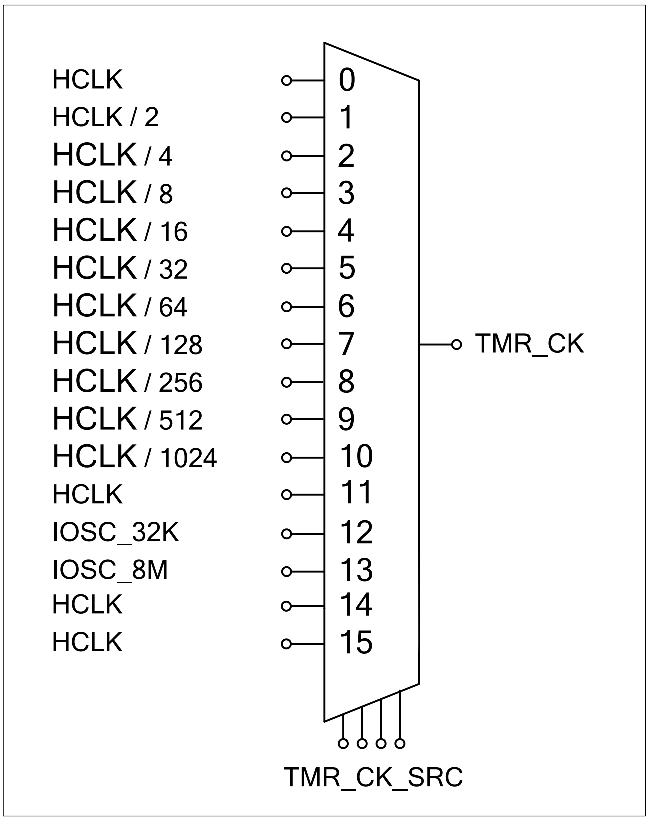
Clock Selection

The counter clock can be provided by the following clock sources:

- HCLK
- IOSC_8M
- IOSC_32K

Internal clock source

A TMx supports 13 internal clock sources. They consist of HCLK / 1~1024, internal 32K oscillator and internal 8M oscillator.



Internal clock source

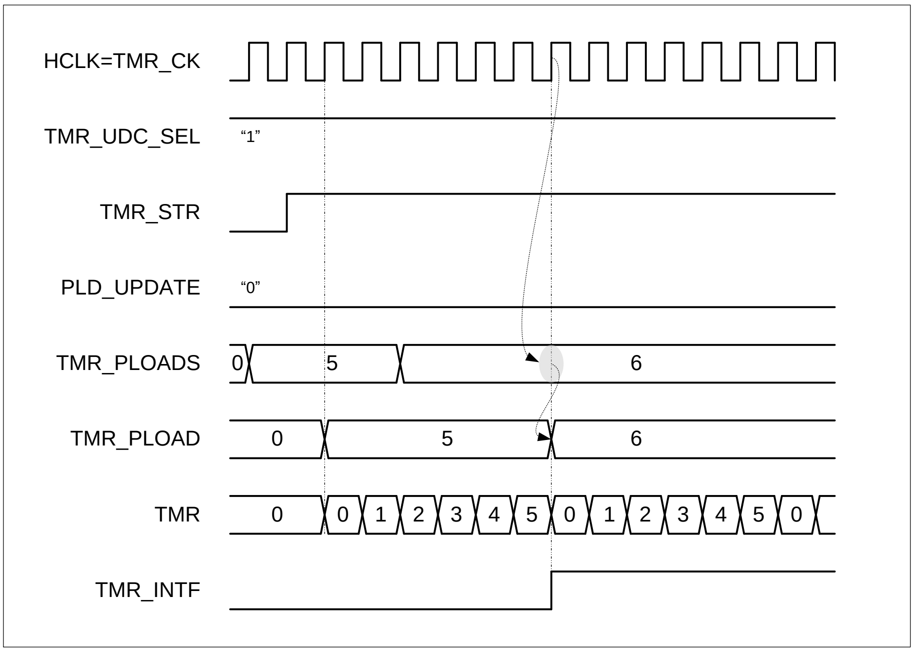
Counter Mode

Up-counting mode

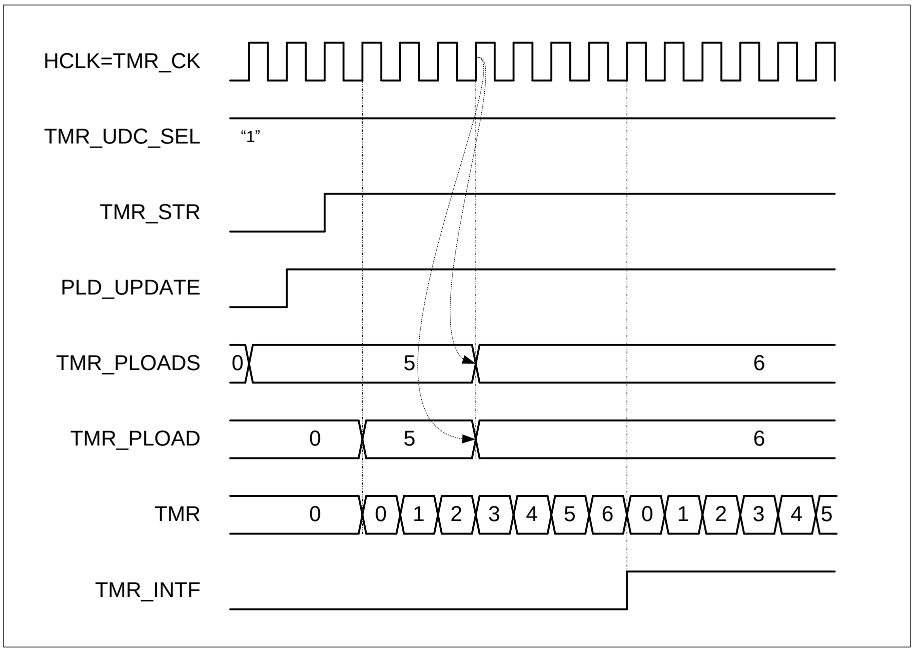
In up-counting mode, the counter counts from 0 to the setting value of **TMx_TMR_PLOAD.TMR_PLOAD**, then timer restarts from 0 and generates a counter overflow event. When **TMx_CTRL.TMR_INTE** is set to 1, the TMx sends an Interrupt event to the CPU in addition to the overflow event.

If the user wants to update the **TMx_TMR_PLOAD.TMR_PLOAD** data, it can be updated in two ways. First, when **TMx_CTRL.PLD_UPDATE** is set to 0, **TMx_TMR_PLOAD.TMR_PLOAD** is updated to **TMx_TMR_PLOAD.TMR_PLOADS** at timer overflow. **TMx_TMR_PLOAD.TMR_PLOAD** and **TMx_TMR_PLOAD.TMR_PLOADS** are updated at the same time when the CPU writes the update data if **TMx_CTRL.PLD_UPDATE** is set to 1.

The following figures show some examples of the counter behavior when **TMx_TMR_PLOAD.TMR_PLOAD**=0x5 and 0x6.



Up-counting timing diagram with **TMx_CTRL.PLD_UPDATE**=0



Up-counting timing diagram with **TMx_CTRL.PLD_UPDATE**=1

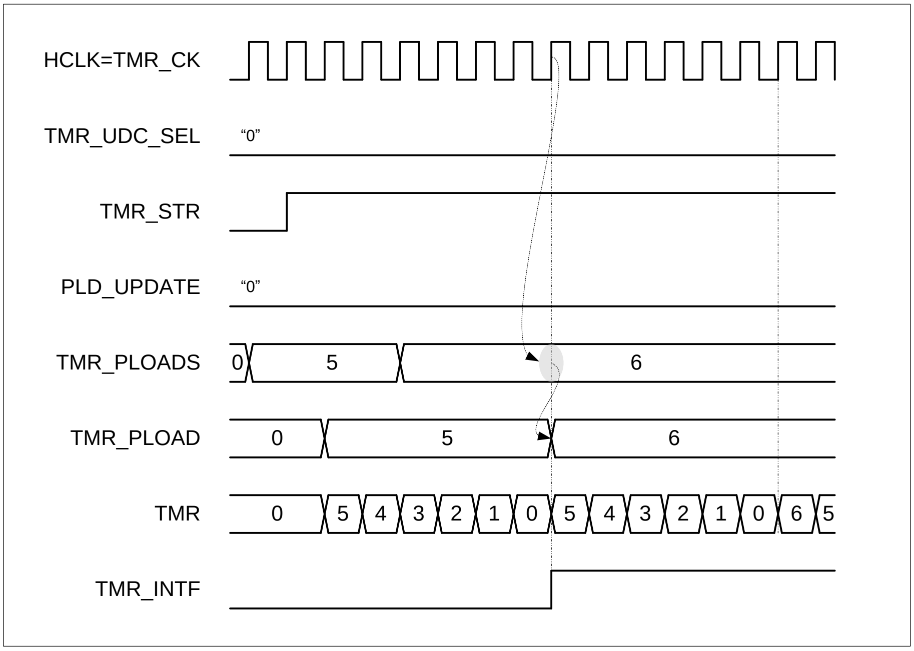
Note: In counting mode, the timer slice of TMx only supports edge alignment mode.

Down-counting mode

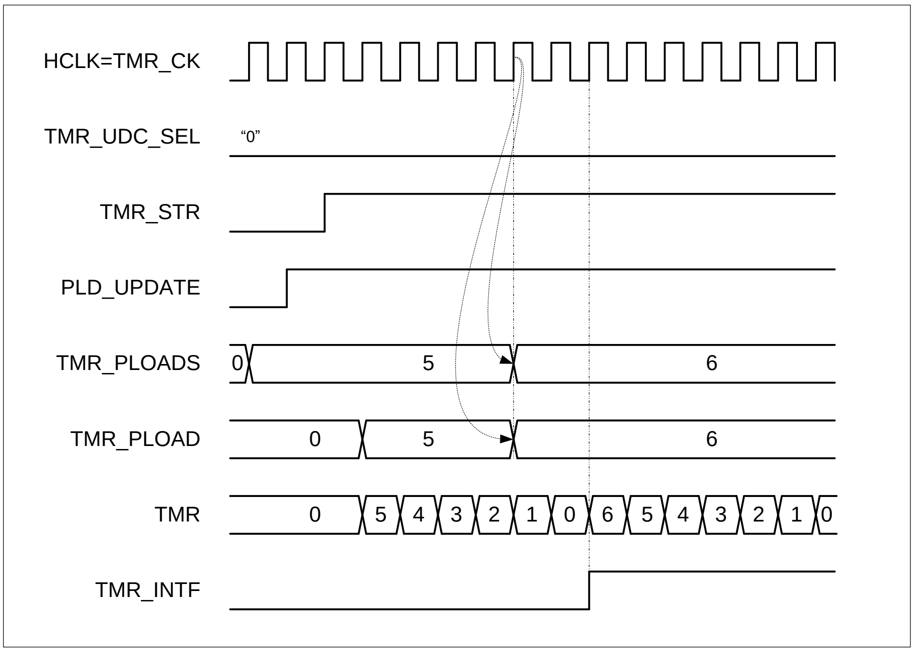
In down-counting mode, the counter counts from the setting value of **TMx_TMR_PLOAD.TMR_PLOAD** to 0, then timer restarts from **TMx_TMR_PLOAD.TMR_PLOAD** and generates a counter underflow event. When **TMx_CTRL.TMR_INTE** is set to 1, the TMx sends an Interrupt event to the CPU in addition to the underflow event.

If the user wants to update the **TMx_TMR_PLOAD.TMR_PLOAD** data, it can be updated in two ways. First, when **TMx_CTRL.PLD_UPDATE** is set to 0, **TMx_TMR_PLOAD.TMR_PLOAD** is updated to **TMx_TMR_PLOAD.TMR_PLOADS** at timer underflow. **TMx_TMR_PLOAD.TMR_PLOAD** and **TMx_TMR_PLOAD.TMR_PLOADS** are updated at the same time when the CPU writes the update data if **TMx_CTRL.PLD_UPDATE** is set to 1.

The following figures show some examples of the counter behavior when **TMx_TMR_PLOAD.TMR_PLOAD**=0x5 and 0x6.



Down-counting timing diagram with **TMx_CTRL.PLD_UPDATE** =0

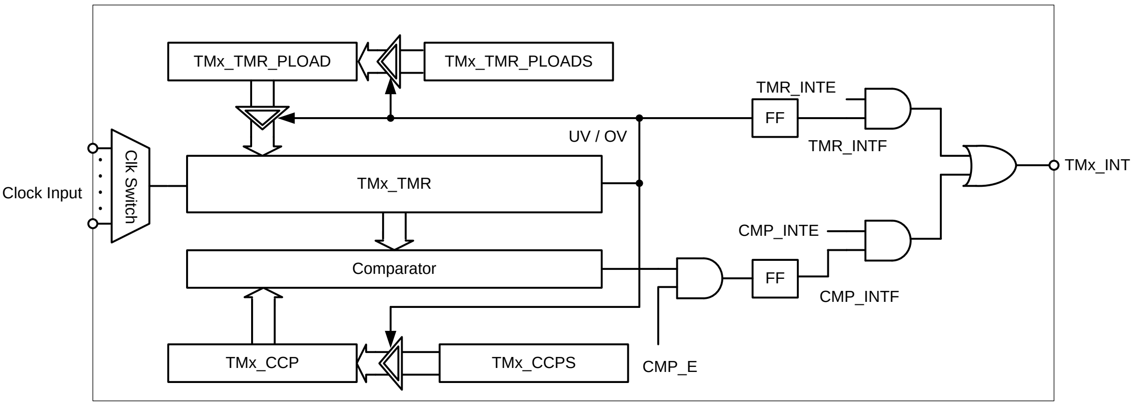


Down-counting timing diagram with **TMx_CTRL.PLD_UPDATE** =1

Note: In counting mode, the timer slice of TMx only supports edge alignment mode.

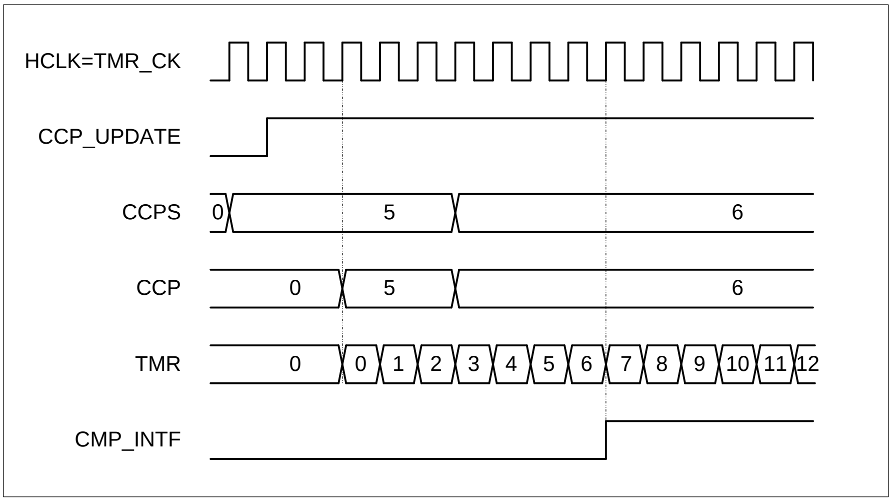
Compare Mode

Each TMx supports 1 compare points (**TMx_CCP.CCP**) to compare mode. This compare point has an enable signal to control the function on and off. In addition, it also has an interrupt enable and interrupt flag. When the value of timer is the same as **TMx_CCP.CCP**, **TMx_STS.CMP_INTF** will be set to 1. If **TMx_CTRL.CMP_INTE** of relative channel is set to 1, TMx will issue an interrupt event to CPU. Figure 10-9 is a block diagram of compare mode.

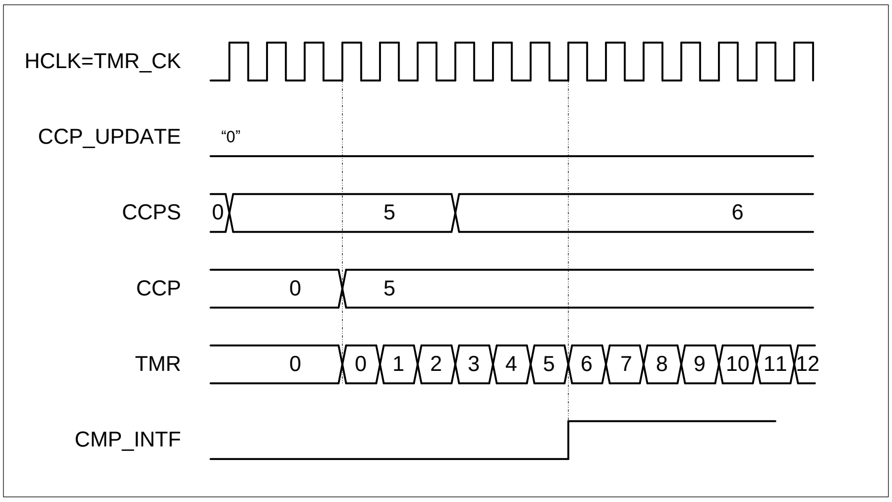


The block diagram of compare mode

The following figures show some examples of the compare mode behavior when **TMx_CCP.CCP**=0x5 and 0x6.



Compare mode timing diagram with **TMx_CTRL.CCP_UPDATE**=0



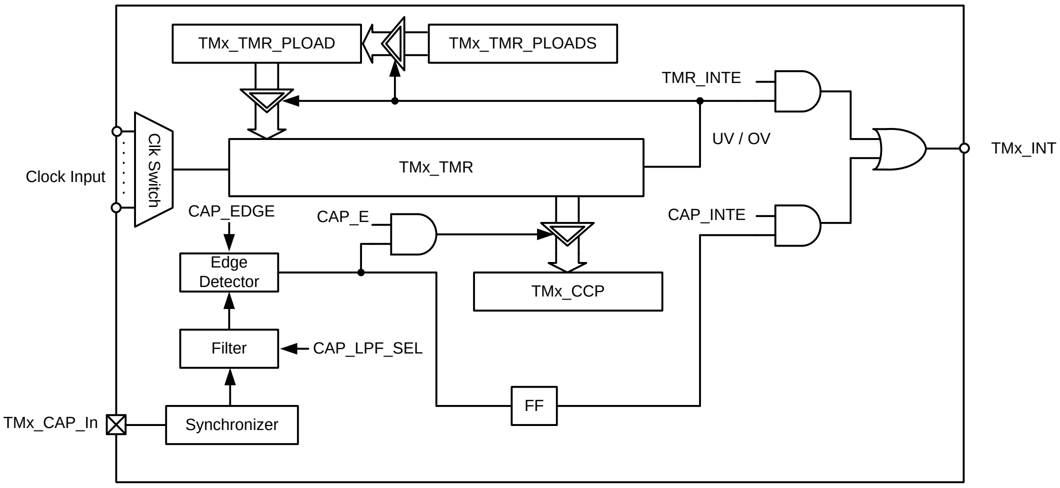
Compare mode timing diagram with **TMx_CTRL.CCP_UPDATE**=1

Capture Mode

In capture mode, each TMx supports 1 capture channels (**TMx_CCP.CCP**) to capture mode. Each capture channel has an enable signal to control the function on and off. In addition, it also has an interrupt enable and interrupt flag.

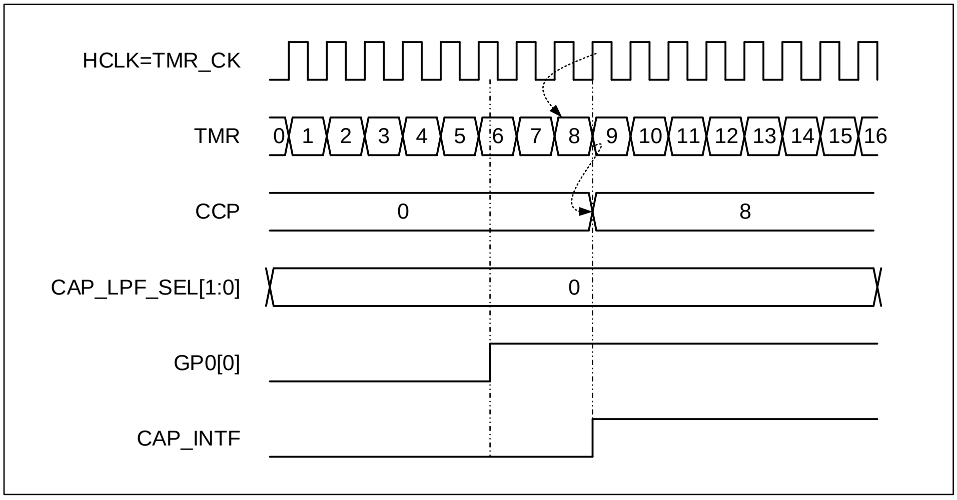
Figure 10-12 give an overview of one capture channel. The capture registers (**TMx_CCP**) are used to latch the value of the counter after a transition detected by the corresponding GPx signal. When a capture occurs, the corresponding **TMx_STS.CAP_INTF** flag is set and an interrupt can be sent if they are enabled.

There are some cases where the user must pay special attention. If a capture occurs while the **TMx_STS.CAP_INTF** flag is already high, then the latest data will over write original data. In addition, **TMx_STS.CAP_INTF** can be cleared by software by writing it to ‘1’.

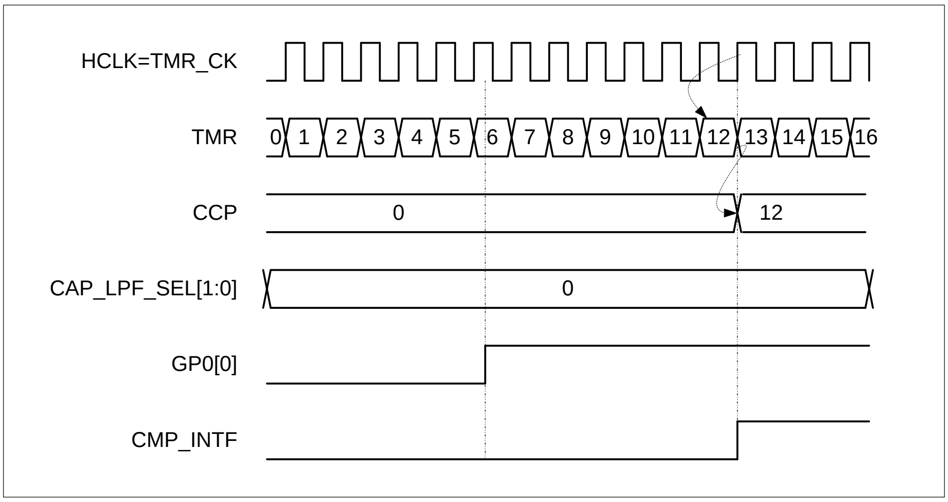


The block diagram of capture mode

Shows how to capture the counter value in **TMx_CCP.CCP** when GP0[0] input rises.



Capture mode timing diagram with **TMx_CTRL.CAP_LPF_SEL[1:0]**=0



Capture mode timing diagram with **TMx_CTRL.CAP_LPF_SEL[1:0]**=1

Register map

Base Address : 0x5005_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
TM0_CTRL	Timer Control Register	0x000	R/W	0x0000 0000
TM0_STS	Timer Status Register	0x004	R/W	0x0000 0000
TM0_TMR_PLOADS	Timer Pre-Load Data Shadow Register	0x008	R/W	0x0000 0000
TM0_TMR_PLOAD	Timer Pre-Load Data Register	0x00C	R	0x0000 0000
TM0_TMR	Timer Actual Value Register	0x010	R	0x0000 0000
TM0_CCPS	Compare / Capture Value Shadow Register	0x014	R/W	0x0000 0000
TM0_CCAPV0	Compare / Capture Buffer 0 Value Register	0x018	R	0x0000 0000
TM0_CAPV1	Capture Buffer 1 Value Register	0x01C	R	0x0000 0000
TM0_CAPV2	Capture Buffer 2 Value Register	0x020	R	0x0000 0000
TM0_CAPV3	Capture Buffer 3 Value Register	0x024	R	0x0000 0000
TM0_MAVG	Capture Buffer Average Register	0x028	R	0x0000 0000

Base Address : 0x5005_1000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
TM1_CTRL	Timer Control Register	0x000	R/W	0x0000 0000
TM1_STS	Timer Status Register	0x004	R/W	0x0000 0000
TM1_TMR_PLOADS	Timer Pre-Load Data Shadow Register	0x008	R/W	0x0000 0000
TM1_TMR_PLOAD	Timer Pre-Load Data Register	0x00C	R	0x0000 0000
TM1_TMR	Timer Actual Value Register	0x010	R	0x0000 0000
TM1_CCPS	Compare / Capture Value Shadow Register	0x014	R/W	0x0000 0000
TM1_CCAPV0	Compare / Capture Buffer 0 Value Register	0x018	R	0x0000 0000
TM1_CAPV1	Capture Buffer 1 Value Register	0x01C	R	0x0000 0000
TM1_CAPV2	Capture Buffer 2 Value Register	0x020	R	0x0000 0000
TM1_CAPV3	Capture Buffer 3 Value Register	0x024	R	0x0000 0000
TM1_MAVG	Capture Buffer Average Register	0x028	R	0x0000 0000

Base Address : 0x5005_2000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
TM2_CTRL	Timer Control Register	0x000	R/W	0x0000 0000
TM2_STS	Timer Status Register	0x004	R/W	0x0000 0000
TM2_TMR_PLOADS	Timer Pre-Load Data Shadow Register	0x008	R/W	0x0000 0000
TM2_TMR_PLOAD	Timer Pre-Load Data Register	0x00C	R	0x0000 0000
TM2_TMR	Timer Actual Value Register	0x010	R	0x0000 0000
TM2_CMPS	Compare Value Shadow Register	0x014	R/W	0x0000 0000
TM2_CMP	Compare Value Register	0x018	R	0x0000 0000

Registers

Timer0/1 Control Register

TM0_CTRL **Timer Control Register** **Address : 0x5005 0000**

TM1_CTRL **Timer Control Register** **Address : 0x5005 1000**

31	30	29	28	27	26	25	24
CAPINT_MODE	CAP_EDGE		CAP_LPF_SEL			CAP_INTE	CAP_E
23	22	21	20	19	18	17	16
CAP_BUF_SEL		CCP_UPDATE	PLD_UPDATE	--		CMP_INTE	CMP_E
15	14	13	12	11	10	9	8
TMR_UDC_SEL	TMR_RST_SEL	TMR_MODE	TMR_INTE	TMR_CK_SRC			
7	6	5	4	3	2	1	0
--							TMR_STR

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[31]</td><td><b>CAPINT_MODE</b></td><td><b>Capture interrupt mode selection bit</b>
0 = Interrupt asserted when every capture signal triggered.
1 = Interrupt asserted when capture buffer index full.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[30]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[30:29]</td><td><b>CAP_EDGE</b></td><td><b>Capture channel sample edge selection bits</b>
<table>
<tr><td>CAP_EDGE</td><td>Capture Edge</td></tr>
<tr><td>00</td><td>NO Action</td></tr>
<tr><td>01</td><td>Rising Edge</td></tr>
<tr><td>10</td><td>Falling Edge</td></tr>
<tr><td>11</td><td>Both Edge</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[28:26]</td><td><b>CAP_LPF_SEL</b></td><td><b>Low pass filter selection bits</b>
<table>
<tr><td>CAP_LPF_SEL</td><td>Filter Cycles (HCLK)</td></tr>
<tr><td>000</td><td>LPF is disable</td></tr>
<tr><td>001</td><td>4 Clock Cycles</td></tr>
<tr><td>010</td><td>16 Clock Cycles</td></tr>
<tr><td>011</td><td>48 Clock Cycles</td></tr>
<tr><td>100</td><td>60 Clock Cycles</td></tr>
<tr><td>101</td><td>72 Clock Cycles</td></tr>
<tr><td>110</td><td>96 Clock Cycles</td></tr>
<tr><td>111</td><td>120 Clock Cycles</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[25]</td><td><b>CAP_INTE</b></td><td><b>Capture mode interrupt enable bits.</b>
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[24]</td><td><b>CAP_E</b></td><td><b>Capture mode enable bits.</b>
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[23:22]</td><td><b>CAP_BUF_SEL</b></td><td><b>Capture value buffer selection bits.</b>
00 = Store the <b>TMR</b> value to CCAPV0.
01 = Store the <b>TMR</b> value to CCAPV0, CAPV1 by sequence.
10 = Store the <b>TMR</b> value to CCAPV0, CAPV1/2 by sequence
11 = Store the <b>TMR</b> value to CCAPV0, CAPV1/2/3 by sequence</td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[21]</td><td><b>CCP_UPDATE</b></td><td><b>Compare data update policy</b>
0 = The <b>TM0_CCP.CCP</b> will be updated while timer is started or timer over/under flow occurred.
1 = The <b>TM0_CCP.CCP</b> will be updated immediately while CPU write data into <b>TM0_CCP.CCP</b>.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[20]</td><td><b>PLD_UPDATE</b></td><td><b>Timer pre-load data update policy</b>
0 = The <b>TM0_TMR_PLOAD.TMR_PLOAD</b> will be updated while timer is over/under flow occurred.
1 = The <b>TM0_TMR_PLOAD.TMR_PLOAD</b> will be updated immediately while CPU write data into <b>TM0_TMR_PLOADS.TMR_PLOADS</b>.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[19:18]</td><td><b>--</b></td><td><b>Reserved</b></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[17]</td><td><b>CMP_INTE</b></td><td><b>Compare mode interrupt enable bits.</b>
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[16]</td><td><b>CMP_E</b></td><td><b>Compare mode enable bits.</b>
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[15]</td><td><b>TMR_UDC_SEL</b></td><td><b>Timer up / down count selection bit</b>
0 = down count
1 = up count</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[14]</td><td><b>TMR_RST_SEL</b></td><td><b>Timer value reset scheme selection bit</b>
0 = When TMR value overflow or underflow.
1 = When TMR value overflow or underflow or capture trigger occurs.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[13]</td><td><b>TMR_MODE</b></td><td><b>Timer operating mode selection bits</b>
<table>
<tr><td>TMR_MODE</td><td>Timer Mode</td></tr>
<tr><td>0</td><td>Counter / Compare Mode</td></tr>
<tr><td>1</td><td>Capture Mode</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[12]</td><td><b>TMR_INTE</b></td><td><b>Timer interrupts enable bit</b>
0 = disable
1 = enable
<b>Note</b>: This bit is used to gate interrupt signal that indicates counter is overflow, underflow or greater than <b>TM0_TMR_PLOAD.TMR_PLOAD</b> value.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[11:8]</td><td><b>TMR_CK_SRC[3:0]</b></td><td><b>Timer input clock source selection bits</b>
<table>
<tr><td>TMR_CK_SRC[3:0]</td><td>Clock Source</td></tr>
<tr><td>0000</td><td>HCLK</td></tr>
<tr><td>0001</td><td>HCLK / 2</td></tr>
<tr><td>0010</td><td>HCLK / 4</td></tr>
<tr><td>0011</td><td>HCLK / 8</td></tr>
<tr><td>0100</td><td>HCLK / 16</td></tr>
<tr><td>0101</td><td>HCLK / 32</td></tr>
<tr><td>0110</td><td>HCLK / 64</td></tr>
<tr><td>0111</td><td>HCLK / 128</td></tr>
<tr><td>1000</td><td>HCLK / 256</td></tr>
<tr><td>1001</td><td>HCLK / 512</td></tr>
<tr><td>1010</td><td>HCLK / 1024</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
</table>

Bit	Name	Description	Access	Reset value
		1011 HCLK 1100 IOSC32K 1101 IOSC8M 1110 ~ 1111 HCLK		
[7:1]	--	**Reserved**	R/W	0x00
[0]	**TMR_STR**	**Timer start bit** 0 = disable 1 = enable	R/W	0x0

Timer2 Control Register

TM2_CTRL **Timer Control Register** **Address : 0x5005 2000**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--		**CCP_UPDATE**	**PLD_UPDATE**	**--**		**CMP_INTE**	**CMP_E**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR_UDC_SEL	**--**		**TMR_INTE**	**TMR_CK_SRC**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							**TMR_STR**

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[31:22]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[21]</td><td><b>CCP_UPDATE</b></td><td><b>Compare data update policy</b>
0 = The <b>TM0_CCP.CCP</b> will be updated while timer is started or timer over/under flow occurred.
1 = The <b>TM0_CCP.CCP</b> will be updated immediately while CPU write data into <b>TM0_CCP.CCP</b>.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[20]</td><td><b>PLD_UPDATE</b></td><td><b>Timer pre-load data update policy</b>
0 = The <b>TM0_TMR_PLOAD.TMR_PLOAD</b> will be updated while timer is over/under flow occurred.
1 = The <b>TM0_TMR_PLOAD.TMR_PLOAD</b> will be updated immediately while CPU write data into <b>TM0_TMR_PLOADS.TMR_PLOADS</b>.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[19:18]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[17]</td><td><b>CMP_INTE</b></td><td><b>Compare mode interrupt enable bits.</b>
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[16]</td><td><b>CMP_E</b></td><td><b>Compare mode enable bits.</b>
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[15]</td><td><b>TMR_UDC_SEL</b></td><td><b>Timer up / down count selection bit</b>
0 = down count
1 = up count</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[14:13]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[12]</td><td><b>TMR_INTE</b></td><td><b>Timer interrupts enable bit</b>
0 = disable
1 = enable
<b>Note</b>: This bit is used to gate interrupt signal that indicates counter is overflow, underflow or greater than <b>TM0_TMR_PLOAD.TMR_PLOAD</b> value.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[11:8]</td><td><b>TMR_CK_SRC[3:0]</b></td><td><b>Timer input clock source selection bits</b>
<table>
<tr><td>TMR_CK_SRC[3:0]</td><td>Clock Source</td></tr>
<tr><td>0000</td><td>HCLK</td></tr>
<tr><td>0001</td><td>HCLK / 2</td></tr>
<tr><td>0010</td><td>HCLK / 4</td></tr>
<tr><td>0011</td><td>HCLK / 8</td></tr>
<tr><td>0100</td><td>HCLK / 16</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td></td><td></td><td><table>
<tr><td>0101</td><td>HCLK / 32</td></tr>
<tr><td>0110</td><td>HCLK / 64</td></tr>
<tr><td>0111</td><td>HCLK / 128</td></tr>
<tr><td>1000</td><td>HCLK / 256</td></tr>
<tr><td>1001</td><td>HCLK / 512</td></tr>
<tr><td>1010</td><td>HCLK / 1024</td></tr>
<tr><td>1011</td><td>HCLK</td></tr>
<tr><td>1100</td><td>IOSC32K</td></tr>
<tr><td>1101</td><td>IOSC8M</td></tr>
<tr><td>1110 ~ 1111</td><td>HCLK</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>[7:1]</td><td>--</td><td>**Reserved**</td><td>R</td><td>0x00</td></tr>
<tr><td>[0]</td><td>**TMR_STR**</td><td>**Timer start bit**
0 = disable
1 = enable</td><td>R/W</td><td>0x0</td></tr>
</table>

TimerX Status Register

TM0_STS	**Timer Status Register**	**Address : 0x5005 0004**
TM1_STS	**Timer Status Register**	**Address : 0x5005 1004**
TM2_STS	**Timer Status Register**	**Address : 0x5005 2004**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--					**CAP_INTF**	**CMP_INTF**	**TMR_INTF**

Bit	Name	Description	Access	Reset value
[31:3]	**--**	**Reserved**	R/W	0x00
[2]	**CAP_INTF**	**Capture mode interrupt flag.** read : 0 = no capture event triggered 1 = capture event triggered write : 0 = no effect 1 = clear this bit **Note :** Timer 2 doesn't support this flag.	R/W	0x0
[1]	**CMP_INTF**	**Compare mode interrupt flag.** read : 0 = no countting equvalent triggered 1 = countting equvalent triggered write : 0 = no effect 1 = clear this bit	R/W	0x0
[0]	**TMR_INTF**	**Timer overflow/underflow interrupts flag** read : 0 = idle / busy 1 = timer interrupt triggered write : 0 = no effect 1 = clear this bit **Note**: This bit will be triggered when counter is overflow, underflow or greater than **TM0_TMR_PLOAD.TMR_PLOAD** value.	R/W	0x0

TimerX Pre-Load Data Shadow Register

TM0_TMR_PLOADS **Timer Pre-Load Data Shadow Register** **Address : 0x5005 0008**

TM1_TMR_PLOADS **Timer Pre-Load Data Shadow Register** **Address : 0x5005 1008**

TM2_TMR_PLOADS **Timer Pre-Load Data Shadow Register** **Address : 0x5005 2008**

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
TMR_PLOADS[15:8]							
7	6	5	4	3	2	1	0
TMR_PLOADS[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**TMR_PLOADS**	**Timer pre-load data or PWM period data shadow register**	R/W	0x0000

TimerX Pre-Load Data Register

TM0_TMR_PLOAD	Timer Pre-Load Data Register	Address : 0x5005 000C
TM1_TMR_PLOAD	Timer Pre-Load Data Register	Address : 0x5005 100C
TM2_TMR_PLOAD	Timer Pre-Load Data Register	Address : 0x5005 200C

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR_PLOAD[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR_PLOAD[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**TMR_PLOAD**	**Timer pre-load data or PWM period data register** **Note**: The **TMx_TMR_PLOAD.TMR_PLOAD** will be filled in **TMx_TMR.TMR** when **TMx_TMR.TMR** is underflow and **TMx_CTRL.TMR_UDC_SEL** is set to 1. Simultaneously, **TMx_CTRL.TMR_MODE** is set as counter, compare or capture mode, unless **TMx_CTRL.TMR_ALIGN_TYPE** is set as one shot mode.	R	0x0000

TimerX Counter Value Register

TM0_TMR	**Timer Counter Value Register**	**Address : 0x5005 0010**
TM1_TMR	**Timer Counter Value Register**	**Address : 0x5005 1010**
TM2_TMR	**Timer Counter Value Register**	**Address : 0x5005 2010**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**TMR**	**Timer actual value**	R	0x0000

TimerX Compare / Capture Value Shadow Register

TM0_CCPS	**Compare / Capture Value Shadow Register**	**Address : 0x5005 0014**
TM1_CCPS	**Compare / Capture Value Shadow Register**	**Address : 0x5005 1014**
TM2_CMPS	**Compare Value Shadow Register**	**Address : 0x5005 2014**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CCPS[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CCPS[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**CCPS**	**Compare / Capture value shadow register** **Note**: In center align mode, These bits are employed when the counter is counted down. In addition, edge align mode will be used too. **Note:** Timer 2 doesn't support capture function.	R/W	0x0000

TimerX Compare / Capture Buffer 0 Value Register

TM0_CCAPV0	Compare / Capture Buffer 0 Value Register	Address : 0x5005 0018
TM1_CCAPV0	Compare / Capture Buffer 0 Value Register	Address : 0x5005 1018
TM2_CMP	Compare Value Register	Address : 0x5005 2018

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CCAP0[15:8]/							
7	6	5	4	3	2	1	0
CCAP0[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	CCAP	**Compare / Capture value register** In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch **TMx_TMR.TMR** data. **Note**: In center align mode, These bits are employed when the counter is counted down. In addition, edge align mode will be used too. **Note:** Timer 2 doesn't support capture function.	R	0x0000

TimerX Capture Buffer 1 Value Register

TM0_CAPV1	Capture Buffer 0 Value Register	Address : 0x5005 001C
TM1_CAPV1	Capture Buffer 0 Value Register	Address : 0x5005 101C

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CAPV1[15:8]							
7	6	5	4	3	2	1	0
CAPV1[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**CAPV1**	**Capture value register** In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch **TMx_TMR.TMR** data. **Note**: In center align mode, These bits are employed when the counter is counted down. In addition, edge align mode will be used too. **Note:** Timer 2 doesn’t support capture function.	R	0x0000

TimerX Capture Buffer 2 Value Register

TM0_CAPV2	Capture Buffer 2 Value Register	Address : 0x5005 0020
TM1_CAPV2	Capture Buffer 2 Value Register	Address : 0x5005 1020

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CAPV2[15:8]							
7	6	5	4	3	2	1	0
CAPV2[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**CAPV2**	**Capture value register** In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch **TMx_TMR.TMR** data. **Note**: In center align mode, These bits are employed when the counter is counted down. In addition, edge align mode will be used too. **Note:** Timer 2 doesn’t support capture function.	R	0x0000

TimerX Capture Buffer 3 Value Register

TM0_CAPV3	Capture Buffer 3 Value Register	Address : 0x5005 0024
TM1_CAPV3	Capture Buffer 3 Value Register	Address : 0x5005 1024

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
CAPV3[15:8]							
7	6	5	4	3	2	1	0
CAPV3[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**CAPV3**	**Capture value register** In compare mode, these bits are employed as compare data. In PWM mode, these bits are employed as PWM duty. In capture mode, these bits are used to latch **TMx_TMR.TMR** data. **Note**: In center align mode, These bits are employed when the counter is counted down. In addition, edge align mode will be used too. **Note:** Timer 2 doesn’t support capture function.	R	0x0000

TimerX Capture Buffer Average Value Register

TM0_MAVG	Capture Buffer Average Value Register	Address : 0x5005 0028
TM1_MAVG	Capture Buffer Average Value Register	Address : 0x5005 1028

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MAVG[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MAVG[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R/W	0x0000
[15:0]	**MAVG**	**Capture Buffer Average value register** $MVAG = \frac{CCAPV0+CAPV1+CAPV2+CAPV3}{4}$	R	0x0000

8.8. General-purpose and Alternate-function I/O (GPIOs and AFIOs)

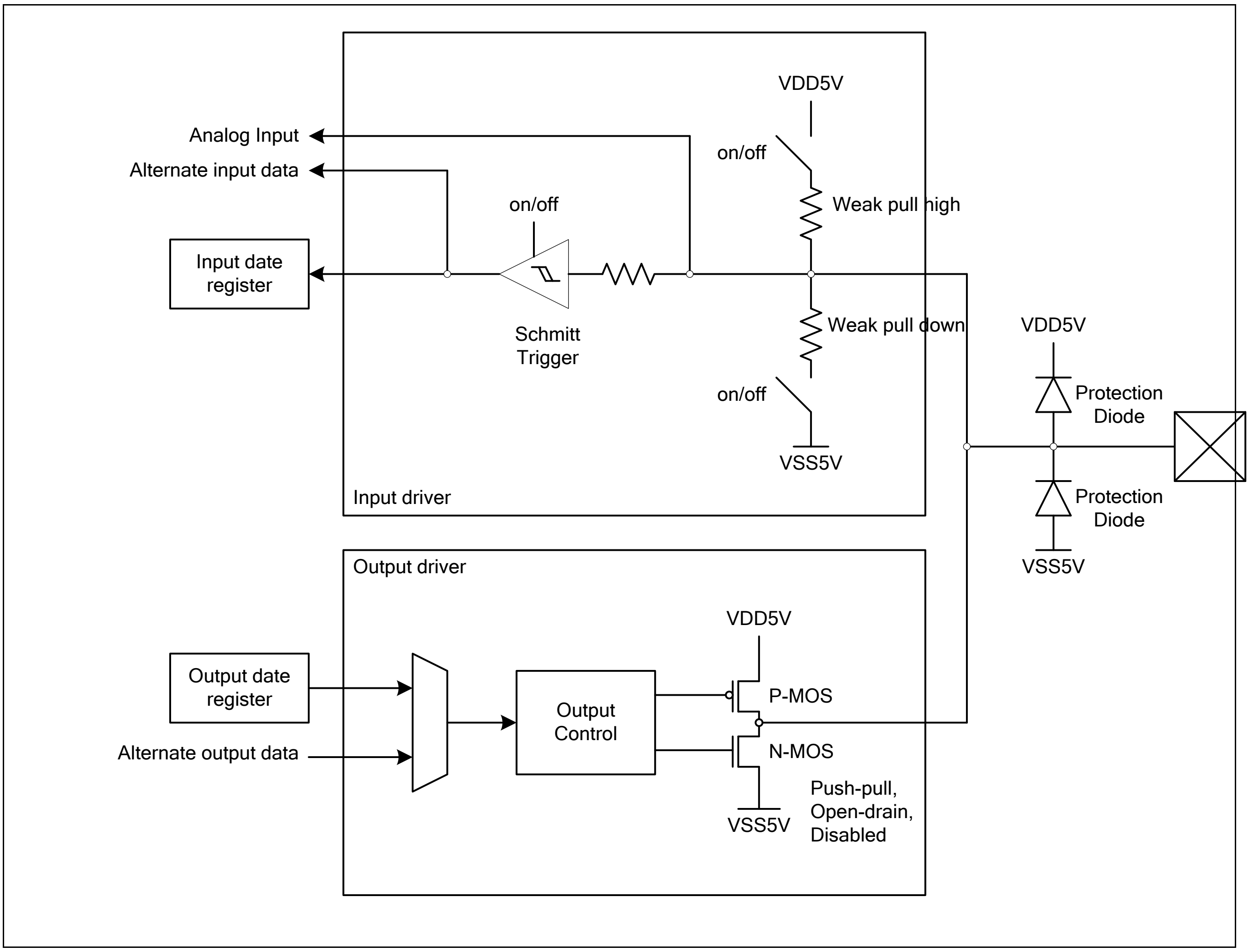
8.8.1. Overview

The GPMQ9103A has up to 5 general purposed I/O ports and it could be shared with other alternated function depending on the IO configuration.

8.8.2. Features

- Each of the general-purpose I/O ports can be individually configured by software in several modes:
  - Input floating
  - Input pull-up
  - Input-pull-down
  - Analog
  - Output open-drain
  - Output push-pull
  - Alternate function open-drain
  - Alternate function push-pull
- Schmitt trigger input
- I/O ports can be configured as interrupt source

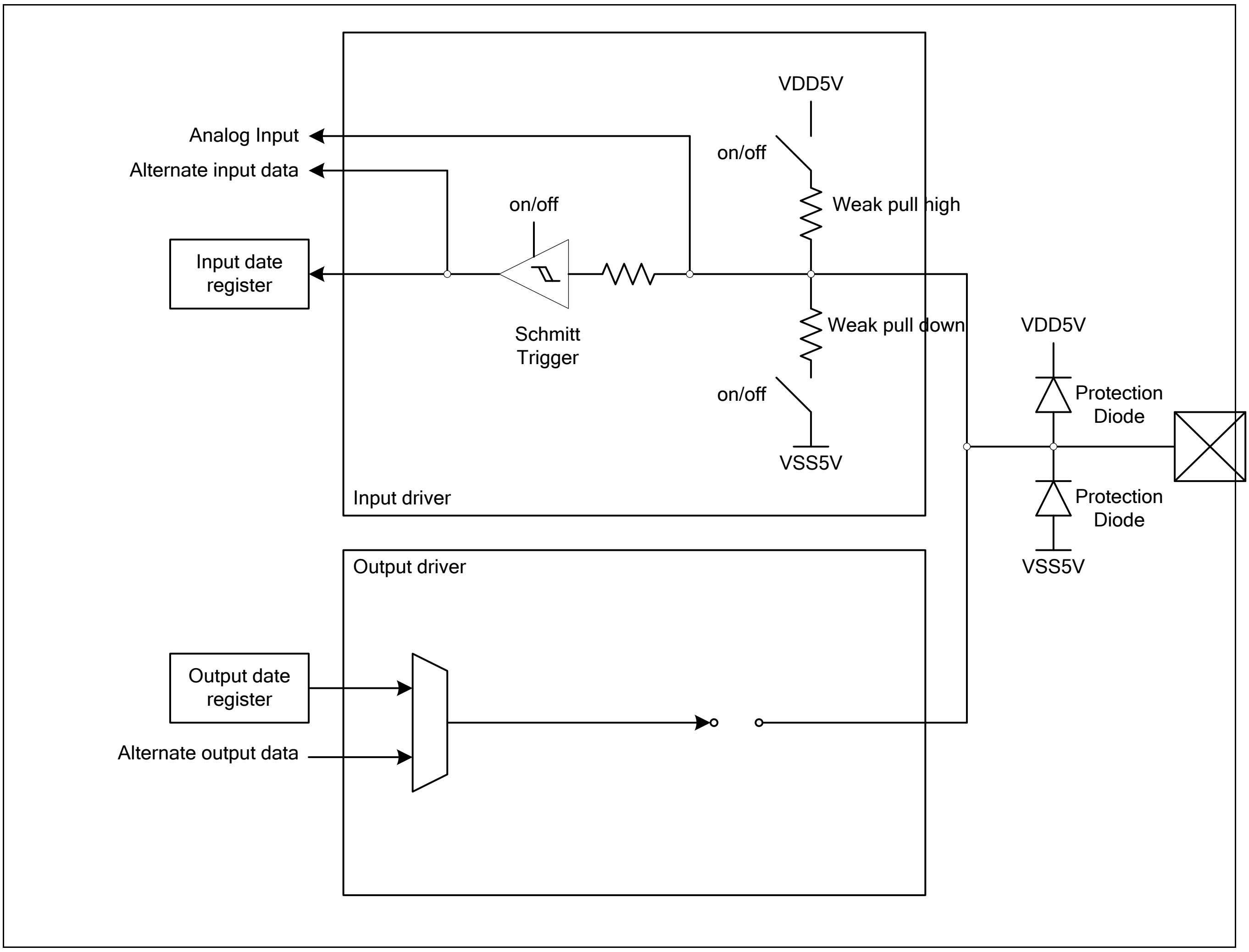
8.8.3. Block Diagram



8.8.4. Input Mode

When the I/O port is configured as Input mode

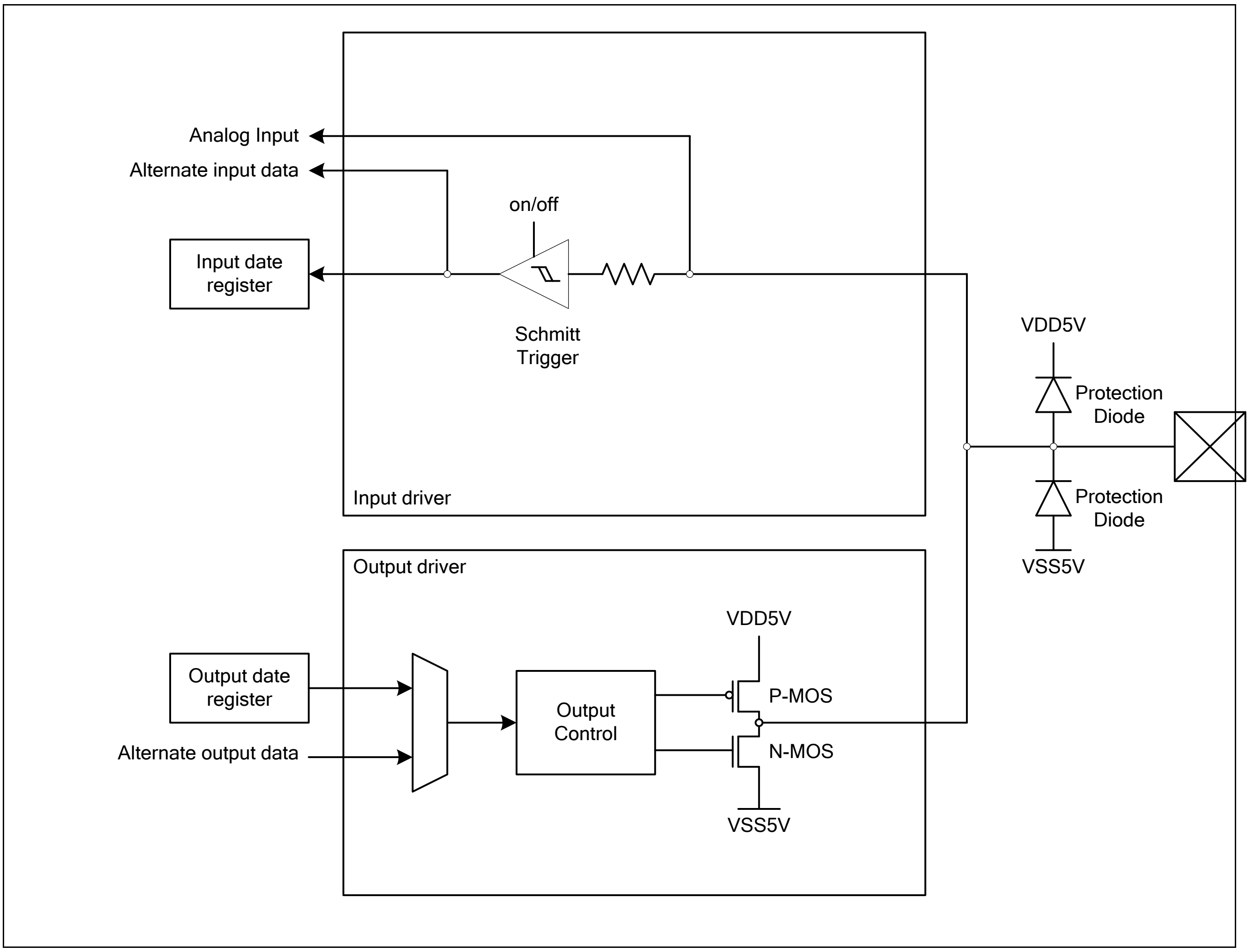
- The output buffer will be disabled
- Schmitt input type support
- The weak pull-up and pull-down resistors are activated or not depending on input configuration (pull-up, pull-down or floating)
- The I/O status in Input data register is sampled by every HCLK cycle.



8.8.5. Output Mode

When the I/O port is configured as output mode

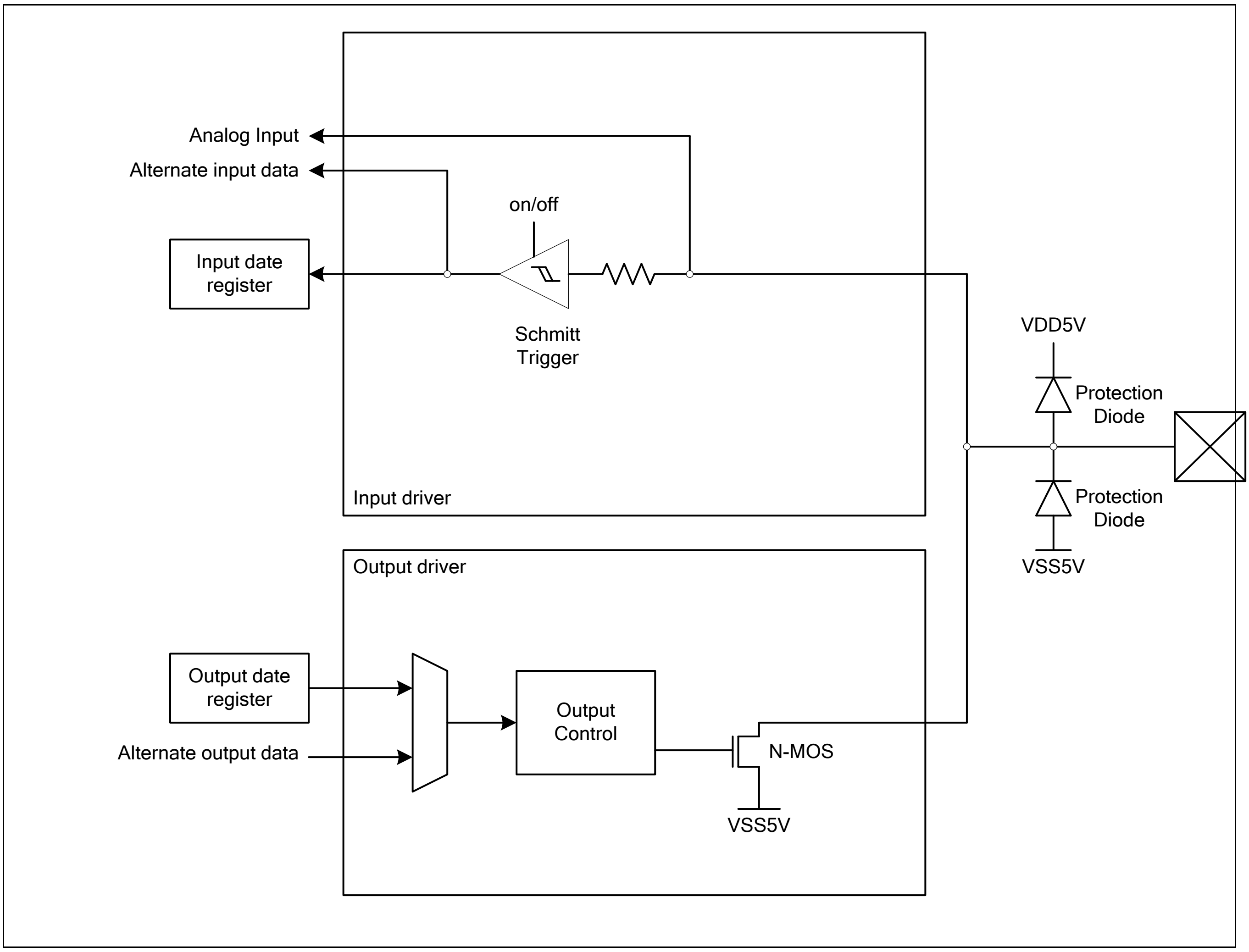
- A ”0” in output data register or alternate output data will make I/O port in low, “1” will make I/O port in high.
- Schmitt input type support
- The weak pull-up and pull-down resistors are disabled.
- The I/O status in Input data register is sampled by every HCLK cycle.



8.8.6. Open Drain Mode

When the I/O port is configured as open drain mode

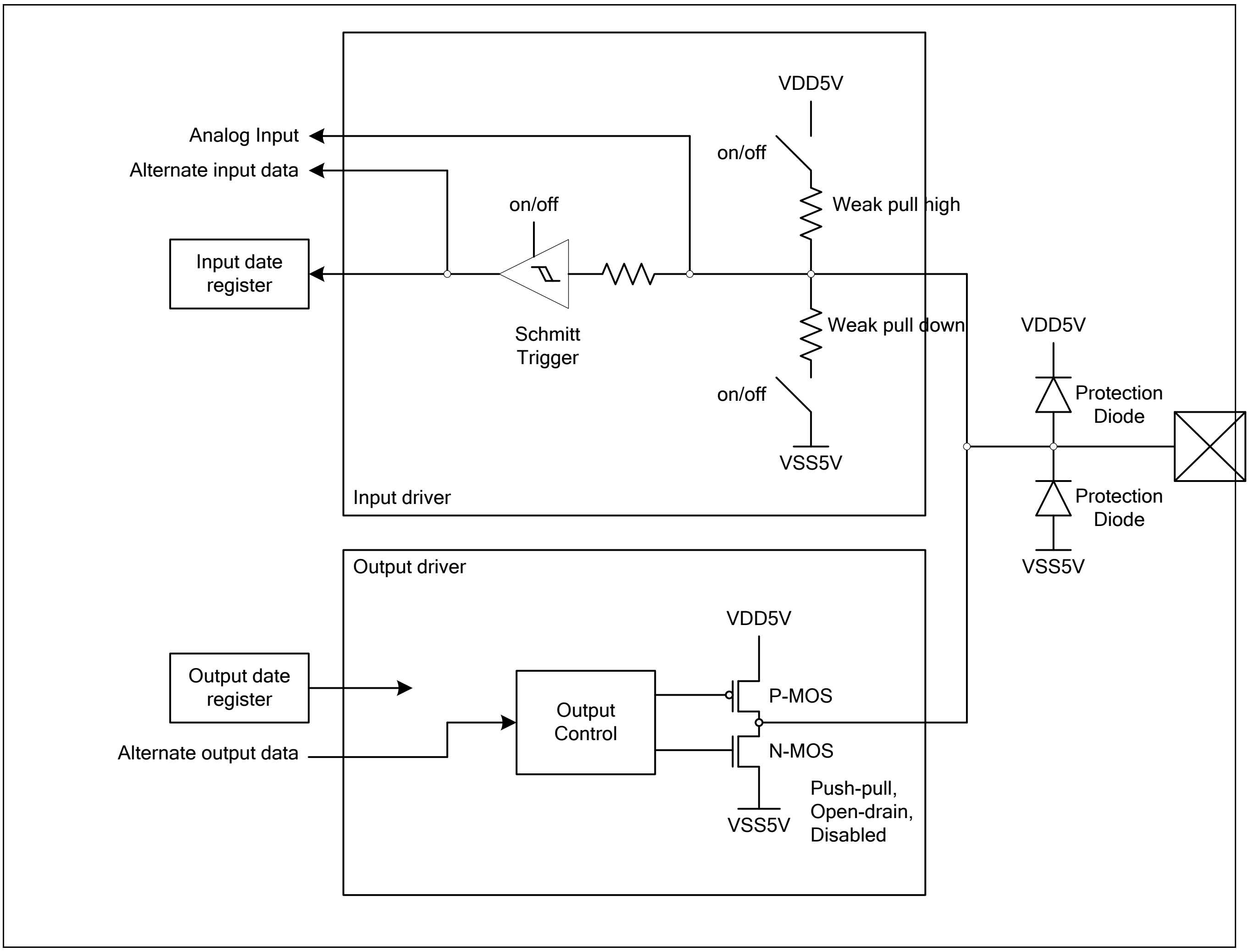
- A ”1” in output data register or alternate output data will make I/O port in low, “0” will make the I/O port in floating.
- Schmitt input type support
- The weak pull-up and pull-down resistors are disabled.
- The I/O status in Input data register is sampled by every HCLK cycle.



8.8.7. Alternate Function Mode

When the I/O port is configured as Alternate function mode

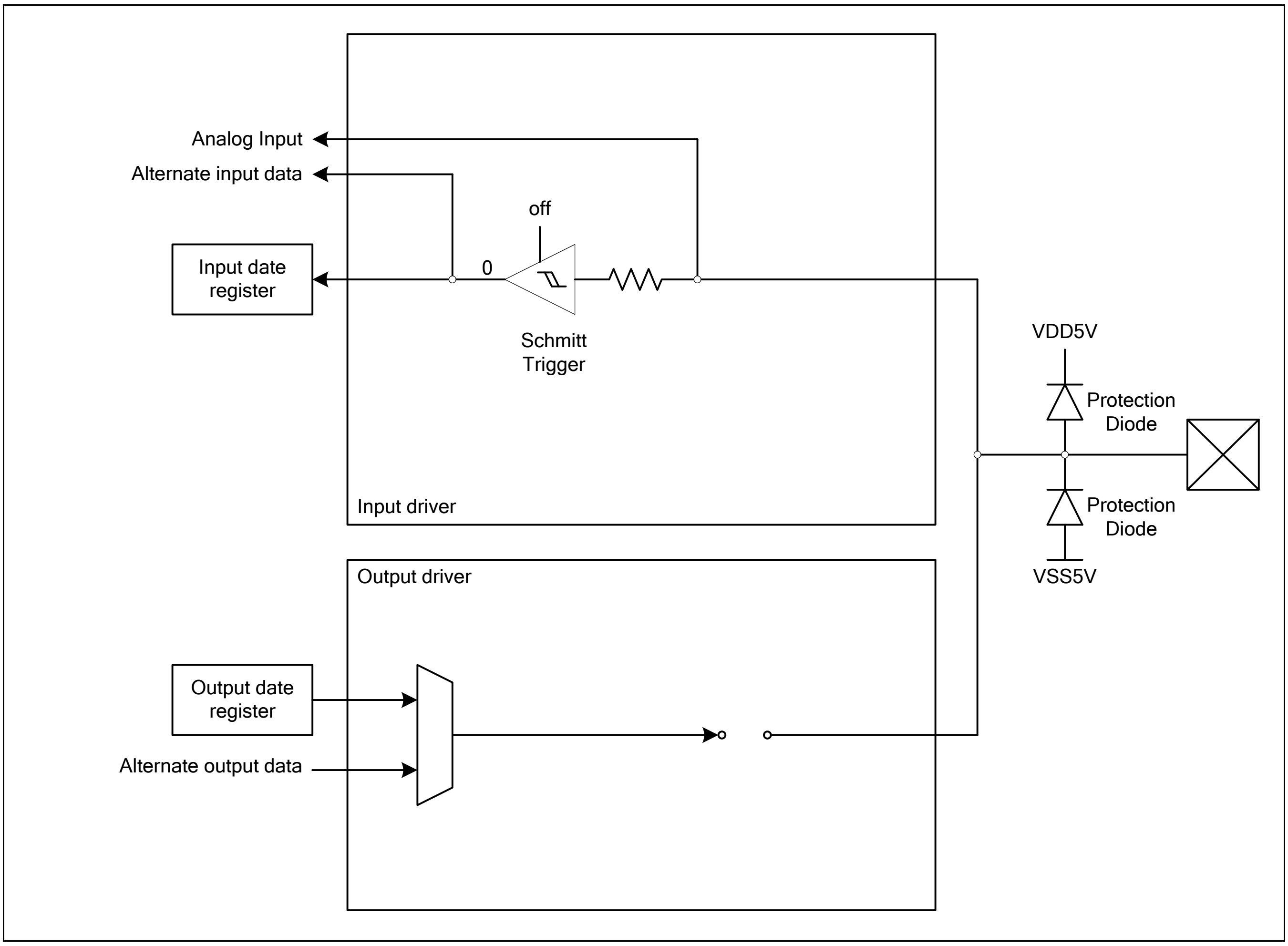
- The output data will be configured to alternate output data.
- I/O port direction configured by alternate function.
- Schmitt input type support
- The weak pull-up and pull-down resistors could be configured by software.
- The I/O status in Input data register is sampled by every HCLK cycle.



8.8.8. Analog Mode

When the I/O port is configured as Alternate function with analog signal mode

- The output buffer will be disabled
- Schmitt are disabled, input data register always read as 0.
- The weak pull-up and pull-down resistors are disabled.



8.8.9. External interrupt

All ports have external interrupt capability. To use external interrupt lines, the port must be configured in input mode.

8.8.10. GPIO mode configuration

Configuration mode		Px_CFG	Px_PULL	Px_IMODE
General purpose output	Push-pull	01	00	--
	Open-drain	10	00	--
Alternate function output	Push-pull	01	00	--
	Open-drain	10	00	--
Input	Input pull-up	00	10	--
	Input pull-down	00	01	--
	Input floating	00	00	--
	Analog	00	00	1x

Register Map

Base Address : 0x5102_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
P0_CFG	GPIO0 Mode Configuration Register	0x000	R/W	0x0000 0000
P0_PULL	GPIO0 Pull-up/Pull-down Resistor Configuration Register	0x004	R/W	0x0000 0000
P0_IMODE	GPIO0 Input Mode Configuration Register	0x008	R/W	0x0000 0000
P0_IDR	GPIO0 Input Data Register	0x010	R	--
P0_ODR	GPIO0 Output Data Register	0x014	R/W	0x0000 0000
P0_ODMSK	GPIO0 Output Data Bit Write Mask Configuration Register	0x018	R/W	0x0000 0000
P0_ALT0	GPIO0 Alternate Mode Configuration Register	0x030	R/W	0x0000 0000
P0n_BCTL n=0…4	GPIO0.n Bit Control Register	0x100+(0X4 * n)	R/W	0x0000 0000

Registers

GPIOx Mode Configuration Register

P0_CFG **GPIO0 Mode Configuration Register** **Address : 0x5102 0000**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--						**Px4_CFG**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Px3_CFG		**Px2_CFG**		**Px1_CFG**		**Px0_CFG**	

Bit	Name	Description	Access	Reset value
[31:10]	**--**	**Reserved**	R	0
[9:8]	**Px4_CFG**	**GPIO Px.4 Mode Configuration** 00 = Input Mode 01 = Push-pull output Mode. 10 = Open Drain Mode 11 = Reserved.	R/W	0
[7:6]	**Px3_CFG**	**GPIO Px.3 Mode Configuration** 00 = Input Mode 01 = Push-pull output Mode. 10 = Open Drain Mode 11 = Reserved.	R/W	0
[5:4]	**Px2_CFG**	**GPIO Px.2 Mode Configuration** 00 = Input Mode 01 = Push-pull output Mode. 10 = Open Drain Mode 11 = Reserved.	R/W	0
[3:2]	**Px1_CFG**	**GPIO Px.1 Mode Configuration** 00 = Input Mode 01 = Push-pull output Mode. 10 = Open Drain Mode 11 = Reserved.	R/W	0
[1:0]	**Px0_CFG**	**GPIO Px.0 Mode Configuration** 00 = Input Mode 01 = Push-pull output Mode. 10 = Open Drain Mode 11 = Reserved.	R/W	0

GPIOx Pull-up/Pull-down Resistor Configuration Register

P0_PULL **GPIO0 Pull-up/Pull-down Resistor Configuration Register** **Address : 0x5102 0004**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--						Px4_PULL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Px3_PULL		Px2_PULL		Px1_PULL		Px0_PULL	

Bit	Name	Description	Access	Reset value
[31:10]	--	**Reserved**	R	0
[9:8]	**Px4_PULL**	**GPIO Px.4 Pull-up/Pull-down Resistor Configuration** 00 = Floating 01 = Pull-Down. 10 = Pull-Up 11 = Reserved.	R/W	0
[7:6]	**Px3_PULL**	**GPIO Px.3 Pull-up/Pull-down Resistor Configuration** 00 = Floating 01 = Pull-Down. 10 = Pull-Up 11 = Reserved..	R/W	0
[5:4]	**Px2_PULL**	**GPIO Px.2 Pull-up/Pull-down Resistor Configuration** 00 = Floating 01 = Pull-Down. 10 = Pull-Up 11 = Reserved.	R/W	0
[3:2]	**Px1_PULL**	**GPIO Px.1 Pull-up/Pull-down Resistor Configuration** 00 = Floating 01 = Pull-Down. 10 = Pull-Up 11 = Reserved.	R/W	0
[1:0]	**Px0_PULL**	**GPIO Px.0 Pull-up/Pull-down Resistor Configuration** 00 = Floating 01 = Pull-Down. 10 = Pull-Up 11 = Reserved..	R/W	0

GPIOx Input Mode Configuration Register

P0_IMODE **GPIO0 Input Mode Configuration Register** **Address : 0x5102 0008**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--						Px4_IMODE	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Px3_IMODE		Px2_IMODE		Px1_IMODE		Px0_IMODE	

Bit	Name	Description	Access	Reset value
[31:10]	--	**Reserved**	R	0
[9:8]	**Px4_IMODE**	**GPIO Px.4 Input Mode Configuration** 00 = Schmitt Trigger input 1x = Input Disable, IDR always read 0.	R/W	0
[7:6]	**Px3_IMODE**	**GPIO Px.3 Input Mode Configuration** 00 = Schmitt Trigger input 1x = Input Disable, IDR always read 0.	R/W	0
[5:4]	**Px2_IMODE**	**GPIO Px.2 Input Mode Configuration** 00 = Schmitt Trigger input 1x = Input Disable, IDR always read 0.	R/W	0
[3:2]	**Px1_IMODE**	**GPIO Px.1 Input Mode Configuration** 00 = Schmitt Trigger input 1x = Input Disable, IDR always read 0.	R/W	0
[1:0]	**Px0_IMODE**	**GPIO Px.0 Input Mode Configuration** 00 = Schmitt Trigger input 1x = Input Disable, IDR always read 0.	R/W	0

GPIOx Input Data Register

P0_IDR **GPIO0 Input Data Register** **Address : 0x5102 0010**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--			Px4_INV	Px3_INV	Px2_INV	Px1_INV	Px0_INV
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			Px4_IDR	Px3_IDR	Px2_IDR	Px1_IDR	Px0_IDR

Bit	Name	Description	Access	Reset value
[31:21]	--	**Reserved**	R	0
[20]	**Px4_INV**	**GPIO Px.4 Input Inversed** 0 = Px4_IDR equals to the corresponding port status 1 = Px4_IDR inverse the corresponding port status	R/W	0
[19]	**Px3_INV**	**GPIO Px.3 Input Inversed** 0 = Px3_IDR equals to the corresponding port status 1 = Px3_IDR inverse the corresponding port status	R/W	0
[18]	**Px2_INV**	**GPIO Px.2 Input Inversed** 0 = Px2_IDR equals to the corresponding port status 1 = Px2_IDR inverse the corresponding port status	R/W	0
[17]	**Px1_INV**	**GPIO Px.1 Input Inversed** 0 = Px1_IDR equals to the corresponding port status 1 = Px1_IDR inverse the corresponding port status	R/W	0
[16]	**Px0_INV**	**GPIO Px.0 Input Inversed** 0 = Px0_IDR equals to the corresponding port status 1 = Px0_IDR inverse the corresponding port status	R/W	0
[15:5]	--	**Reserved**	R	0
[4]	**Px4_IDR**	**GPIO Px.4 Input Data** This bit is read only and the bit contain reflect or inverse the port status.	R	--
[3]	**Px3_IDR**	**GPIO Px.3 Input Data** This bit is read only and the bit contain reflect or inverse the port status.	R	--
[2]	**Px2_IDR**	**GPIO Px.2 Input Data** This bit is read only and the bit contain reflect or inverse the port status.	R	--
[1]	**Px1_IDR**	**GPIO Px.1 Input Data** This bit is read only and the bit contain reflect or inverse the port status.	R	--
[0]	**Px0_IDR**	**GPIO Px.0 Input Data** This bit is read only and the bit contain reflect or inverse the port status.	R	--

GPIOx Output Data Register

P0_ODR **GPIO0 Output Data Register** **Address : 0x5102 0014**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--			**Px4_ODR**	**Px3_ODR**	**Px2_ODR**	**Px1_ODR**	**Px0_ODR**

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4]	**Px4_ODR**	**GPIO Px.4 Output Data** 0 = Driver low on port in Push-pull and Open drain mode both. 1 = Driver high on port in Push-pull mode or Hi-z in Open drain mode.	R/W	0
[3]	**Px3_ODR**	**GPIO Px.3 Output Data** 0 = Driver low on port in Push-pull and Open drain mode both. 1 = Driver high on port in Push-pull mode or Hi-z in Open drain mode.	R/W	0
[2]	**Px2_ODR**	**GPIO Px.2 Output Data** 0 = Driver low on port in Push-pull and Open drain mode both. 1 = Driver high on port in Push-pull mode or Hi-z in Open drain mode.	R/W	0
[1]	**Px1_ODR**	**GPIO Px.1 Output Data** 0 = Driver low on port in Push-pull and Open drain mode both. 1 = Driver high on port in Push-pull mode or Hi-z in Open drain mode.	R/W	0
[0]	**Px0_ODR**	**GPIO Px.0 Output Data** 0 = Driver low on port in Push-pull and Open drain mode both. 1 = Driver high on port in Push-pull mode or Hi-z in Open drain mode.	R/W	0

GPIOx Output Data Bit Write Mask Configuration Register

P0_ODMSK		GPIO0 Output Data Bit Write Mask Configuration Register				Address : 0x5102 0018	
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--			Px4_ODMSK	Px3_ODMSK	Px2_ODMSK	Px1_ODMSK	Px0_ODMSK

Bit	Name	Description	Access	Reset value
[31:5]	--	**Reserved**	R	0
[4]	**Px4_ODMSK**	**GPIO Px.4 Output Data Bit Write Mask** 0 = Corresponding Px_ODR bit can be updated. 1 = Corresponding Px_ODR bit cannot be updated.	R/W	0
[3]	**Px3_ODMSK**	**GPIO Px.3 Output Data Bit Write Mask** 0 = Corresponding Px_ODR bit can be updated. 1 = Corresponding Px_ODR bit cannot be updated.	R/W	0
[2]	**Px2_ODMSK**	**GPIO Px.2 Output Data Bit Write Mask** 0 = Corresponding Px_ODR bit can be updated. 1 = Corresponding Px_ODR bit cannot be updated.	R/W	0
[1]	**Px1_ODMSK**	**GPIO Px.1 Output Data Bit Write Mask** 0 = Corresponding Px_ODR bit can be updated. 1 = Corresponding Px_ODR bit cannot be updated.	R/W	0
[0]	**Px0_ODMSK**	**GPIO Px.0 Output Data Bit Write Mask** 0 = Corresponding Px_ODR bit can be updated. 1 = Corresponding Px_ODR bit cannot be updated.	R/W	0

GPIOx Alternate Function Configuration Register

P0_ALT0 **GPIO0 Alternate Function Configuration Register** **Address : 0x5102 0030**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--				**Px4_ ALT**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Px3_ ALT				**Px2_ ALT**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Px1_ ALT				**Px0_ALT**			

Bit	Name	Description	Access	Reset value
[31:20]	**--**	**Reserved**	R	0
[19:16]	**Px4_ ALT**	**GPIO Px.4 Alternate Function Configuration** 0000 = GPIO Others setting to refer alternate function table.	R/W	0
[15:12]	**Px3_ ALT**	**GPIO Px.3 Alternate Function Configuration** 0000 = GPIO Others setting to refer alternate function table.	R/W	0
[11:8]	**Px2_ ALT**	**GPIO Px.2 Alternate Function Configuration** 0000 = GPIO Others setting to refer alternate function table.	R/W	0
[7:4]	**Px1_ ALT**	**GPIO Px.1 Alternate Function Configuration** 0000 = GPIO Others setting to refer alternate function table.	R/W	0
[3:0]	**Px0_ ALT**	**GPIO Px.0 Alternate Function Configuration** 0000 = GPIO Others setting to refer alternate function table.	R/W	0

GPIOx.n Bit Control Register

P0n_BCTL **GPIO0.n Bit Control Register** **Address : 0x5102**

n=0…4 **0100+(0x4*n)**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--							**Pxn_BCTL**

Bit	Name	Description	Access	Reset value
[31:1]	**--**	**Reserved**	R	0
[0]	**Pxn_BCTL**	**GPIO Px.n Bit Control** 0 = Corresponding bit n of Px_ODR will be reset 1 = Corresponding bit n of Px_ODR will be set	R/W	0

8.9. Analog to Digital Converters (ADC) Controller

8.9.1. Overview

The GPMQ9103A incorporates one 10-bit ADC. ADC has up to 12 multiplexed channels allowing it measure 12 signals sources. The functionally of A/D conversion can be performed in regular, regular with loop mode and injection mode. The result of the conversion data is stored in a data register. User can set data alignment with left-aligned or right-aligned.

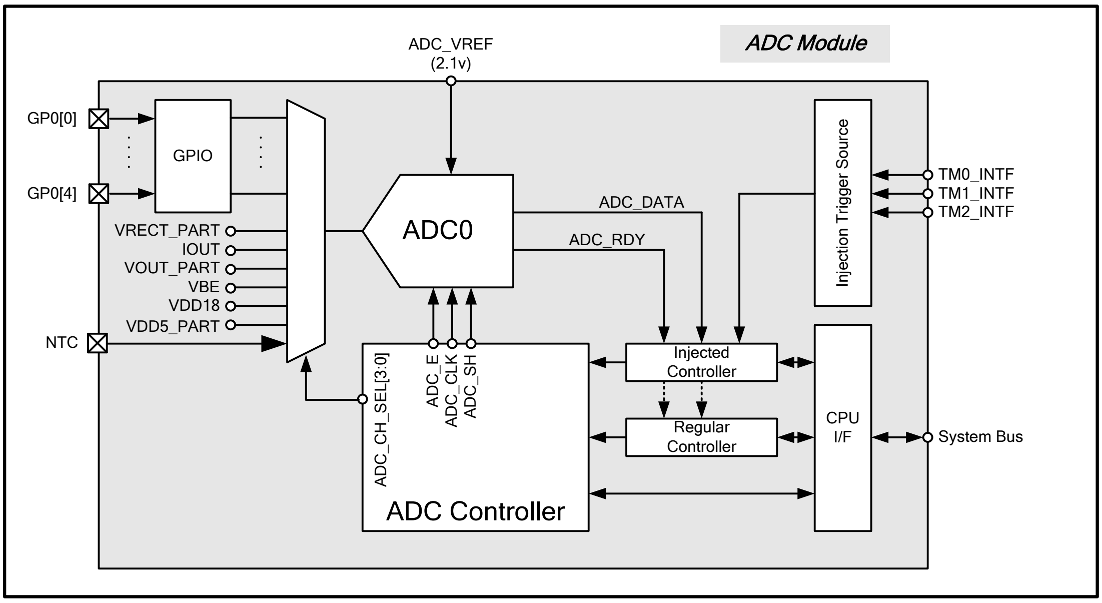
A function of analog watchdog feature allows the application to detect input-voltage. It is used to detect whether the input voltage is out of the user-defined range.

The input clock of ADC is generated from the PCLK clock that fixed in 1.5MHz.

8.9.2. Features

- 10-bit resolution
- ADC has up to 12 channels
- Interrupt generation at end of conversion, end of Injected conversion and analog watchdog event
- Single, regular and regular scan conversion modes
- Scan mode for automatic conversion of channel 0 to channel ‘n’
- Data alignment with left and right alignment
- Channel by channel programmable sampling time
- External trigger option for injected conversion
- ADC supply voltage requirement: 4.5 V
- ADC input range from GPIO/NTC: 0V to VREF+(2.1v)

8.9.3. Block Diagram



ADC block diagram

8.9.4. Function Description

8.9.4.1. ADC on-off control

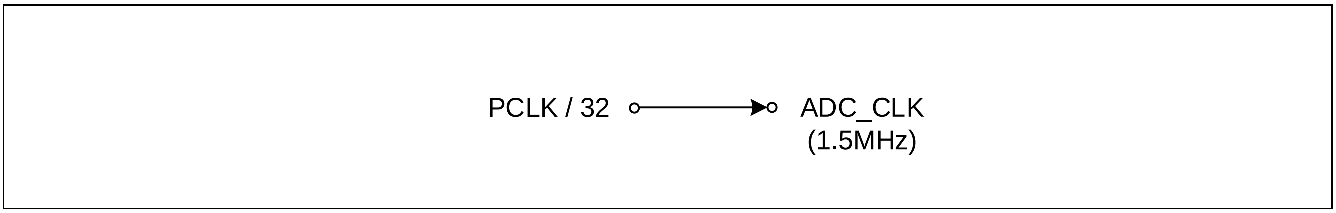
The ADC controller can be turn-on by setting the **ADCx_CTRL0.ADC_E** bit. The user must note that the ADC reference voltage must be set via **ADC_GCTRL** before using the ADC. After **ADCx_CTRL0.ADC_E** set to 1, ADC_RDY is high to indicate ADC is initializing. Until ADC_RDY becomes 0, then the user can start using the ADC.

In regular mode, conversion starts when **ADCx_CTRL0.SFT_STR** bit is set to 1. In injected mode, conversion starts when one of eight trigger source is triggered.

You can stop conversion and put the ADC in power down mode by disabling the **ADCx_CTRL0.ADC_E** bit.

8.9.4.2. ADC Clock

The ADC_CLK that divided by 32 and would be 1.5MHz. The ADC_CLK is synchronous with the PCLK.



ADC_CLK clock source

8.9.4.3. Channel Selection

Each ADC has up to 12 multiplexed channels allowing it measure signals from 6 external and 6 internal sources. It is possible to operate the conversions in two modes. A mode consists of a sequence of conversions which can be done on any channel and in any order.

- The **regular mode** is composed of up to 12 conversions. The regular channels and their order in the conversion sequence must be selected in the **ADCx_REG_SEQ0/1** registers. The total number of conversions in the regular mode must be written in the **ADCx_CTRL1.REG_CH_NUM**.
- The **injected mode** is composed of up to 4 conversions. The injected channels and their order in the conversion sequence must be selected in the **ADCx_INJ_SEQ** register. The total number of conversions in the injected mode must be written in the **ADCx_CTRL1.INJy_E (y=0~3)** bits.

If the **ADCx_REG_SEQ0/1** or **ADCx_CTRL1.INJy_E (y=0~3)** registers are modified during a conversion, the current conversion data will be abnormal.

Table shows the channel mapping of ADCx.

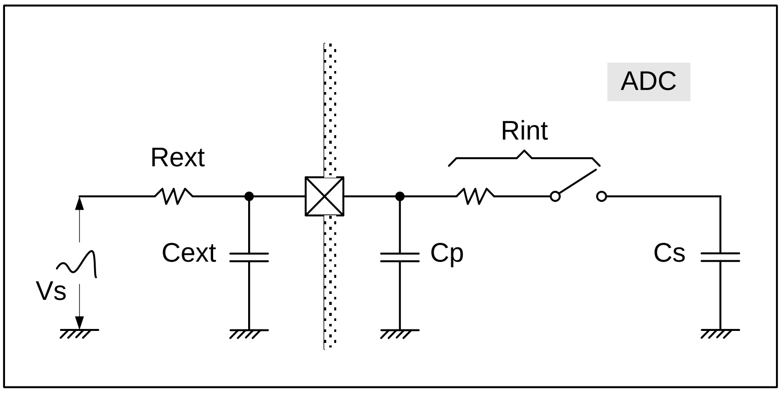
ADCx channel mapping table

	ADC0
Channel #	***Channel Mapping***
0	VRECT_PART
1	IOUT
2	VOUT_PART
3	VDD18
4	DIE_VBE
5	NTC
6	GPIO[0]
7	GPIO[1]
8	GPIO[2]
9	GPIO[3]
10	GPIO[4]
11	VDD5_PART

8.9.4.4. Sampling path

The ADC structure of the GPMQ9103A uses a switched capacitor Cs to save the input signal. During signal sampling, the capacitor C*s* is connected to the analog input via the multiplexer. The parasitic resistor of internal switch will be molding to Rint. An analog voltage source Vs that should be converted. The value of Rext, Cext, Rint and Cs are strongly defining the required length of the sample cycles.

The capacitance value of switched capacitor Cs is approximately 10pf. The parasitic resistance value of Rint is range in 200Ω ~ 14KΩ which is affected by different operating voltage and temperature. According to different application, user should be take care the resistor and capacitor value of Rext and Cext. Figure shows a single channel model of ADC.



A single channel model of ADC.

8.9.4.5. Regular conversion mode

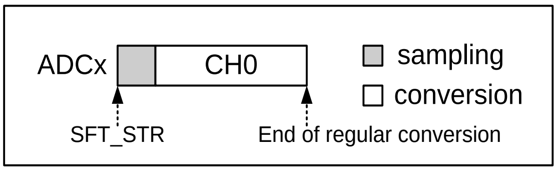
8.9.4.5.1. Regular conversion with loop function disabled

In regular conversion mode, set **ADCx_CTRL1.REG_E = 1,** the ADC does one conversion. This mode is started by setting the **ADCx_CTRL0.SFT_STR** only.

Once the conversion of the selected channel is complete:

- If a regular channel is converted:

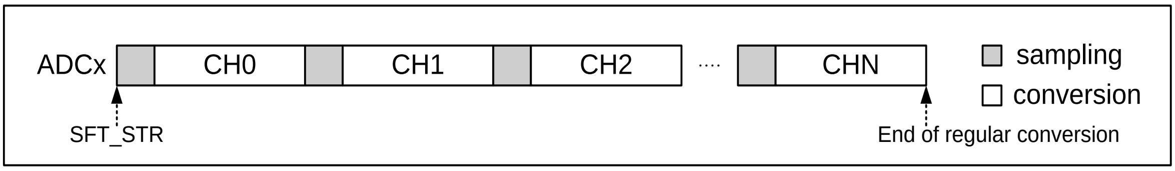
1. The converted data is stored in the 16-bit **ADCx_REG_DAT** register
2. The **ADCx_CTRL0.REG_INTF** flag is set
3. An interrupt is generated if the **ADCx_CTRL0.REG_INTE** is set.
4. User clear **ADCx_CTRL0.REG_INTF** flag by write **ADCx_CTRL0.REG_INTF** to 1.



A regular conversion

- If N regular channel are converted:

1. The converted data is stored in the 16-bit **ADCx_REG_DAT** register
2. The **ADCx_CTRL0.REG_INTF** flag is set
3. An interrupt is generated if the **ADCx_CTRL0.REG_INTE** is set.
4. User clear **ADCx_CTRL0.REG_INTF** flag by write **ADCx_CTRL0.REG_INTF** to 1.
5. Past some ADC_CLK cycles that is set in **ADCx_CTRL1.REG_SEQ_GAP**, do next conversion until all of the regular channel are finished.



N regular conversion

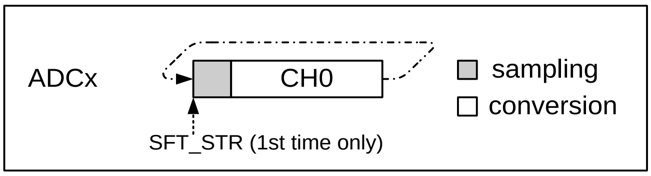
8.9.4.5.2. Regular conversion with loop function enabled

In regular conversion with loop enable mode the ADC will repeat the channel conversion action. This mode is started by setting the **ADCx_CTRL0.SFT_STR**. Before start ADC conversion, user must set **ADCx_CTRL0.LOOP_E** to 1. The data conversion will be triggered by hardware automatically until software disable ADC or set **ADCx_CTRL0.LOOP_E** to 0.

Once the conversion of the selected channel is complete:

- If only one regular channel is selected:

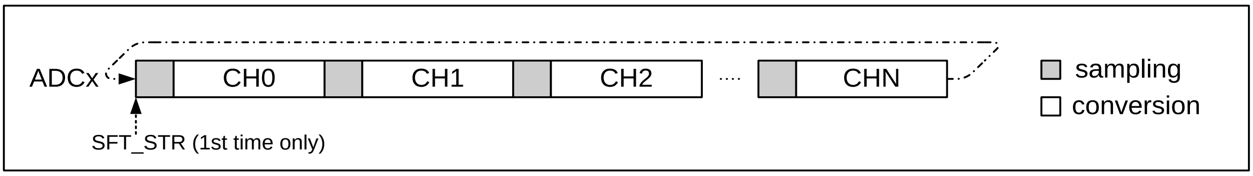
1. The converted data is stored in the 16-bit **ADCx_REG_DAT** register
2. The **ADCx_CTRL0.REG_INTF** flag is set
3. An interrupt is generated if the **ADCx_CTRL0.REG_INTE** is set.
4. User clear **ADCx_CTRL0.REG_INTF** flag by write **ADCx_CTRL0.REG_INTF** to 1.
5. Past some ADC_CLK cycles that are set in **ADCx_CTRL1.REG_SEQ_GAP**, does next iteration until software disable ADC or set **ADCx_CTRL0.LOOP_E** to 0.



A regular conversion with loop function enable

- If N regular channel are converted:

1. The converted data is stored in the 16-bit **ADCx_REG_DAT** register
2. The **ADCx_CTRL0.REG_INTF** flag is set
3. An interrupt is generated if the **ADCx_CTRL0.REG_INTE** is set.
4. User clear **ADCx_CTRL0.REG_INTF** flag by write **ADCx_CTRL0.REG_INTF** to 1.
5. Past some ADC_CLK cycles that are set in **ADCx_CTRL1.REG_SEQ_GAP**, do next conversion until all of the regular channel are finished.
6. Past some ADC_CLK cycles that are set in **ADCx_CTRL1.REG_SEQ_GAP**, does next iteration until software disable ADC or set **ADCx_CTRL0.LOOP_E** to 0.

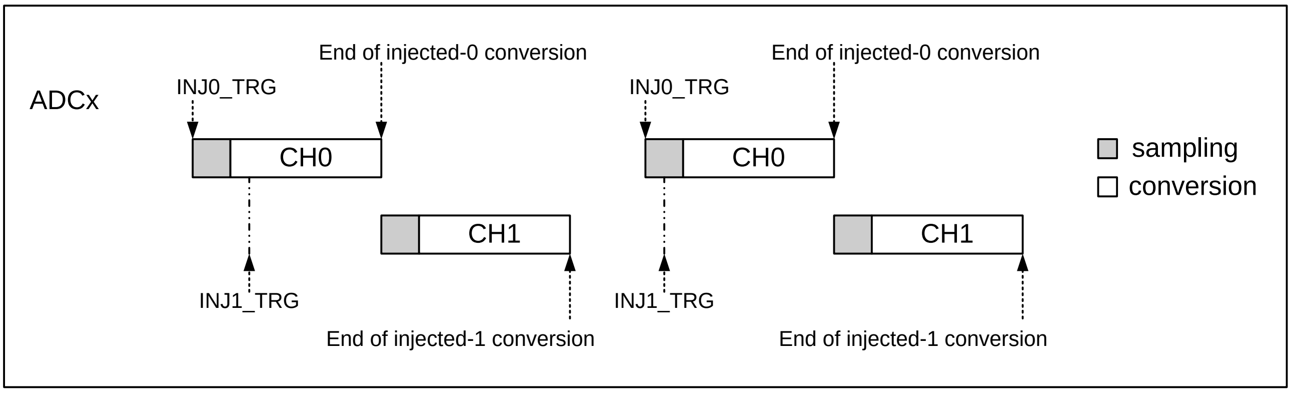


N regular conversion with loop function enable

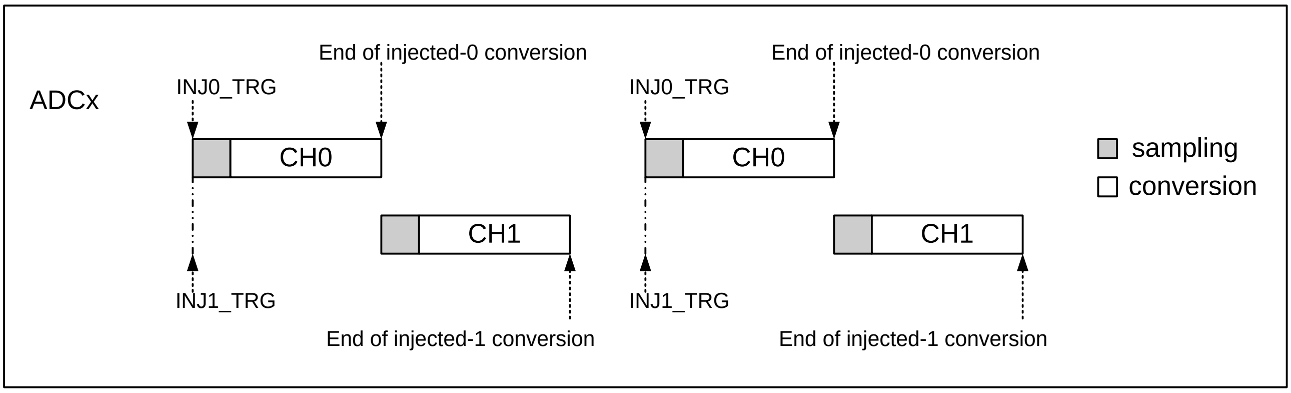
8.9.4.6. Injected conversion mode

To use triggered injection, one of ADCx_CTRL1.INJy_E(y=0 ~ 3) must be set. Each injected channel supports 8 trigger sources. User can select a trigger source by set **ADCx_CTRL1.INJy_TRG_SEL(y=0~3)**.The priority of injected channel is INJ0 > INJ1 > INJ2 > INJ3. In addition, the priority of any injection channel is always higher than the regular channel. Below explains how injected mode works.

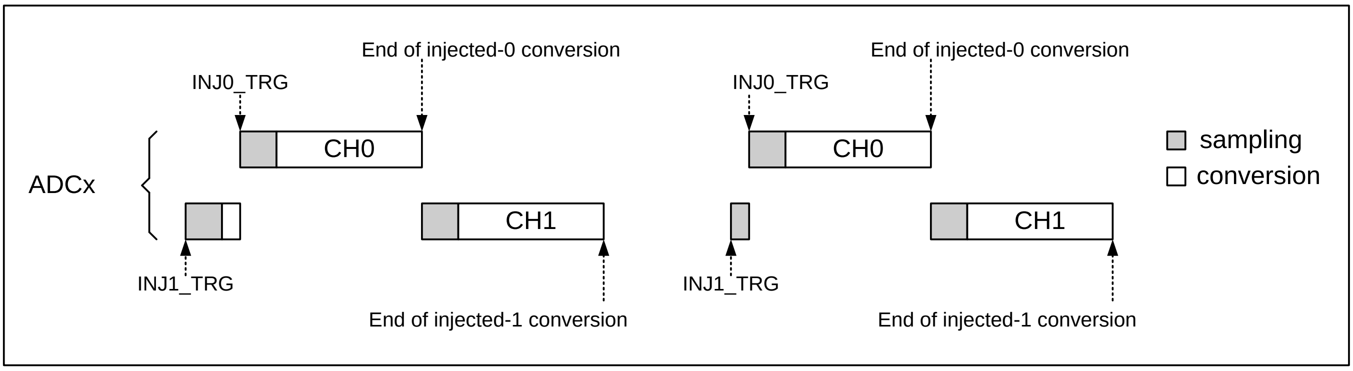
1. Start conversion of an injected channel by trigger source.
2. The converted data is stored in the 16-bit **ADCx_INJy_DAT** register
3. The **ADCx_CTRL0.INJy_INTF** flag is set
4. User clear **ADCx_CTRL0.INJy_INTF** flag by write **ADCx_CTRL0.INJy_INTF** to 1.


ADCx
End of injected-0 conversion
INJ0_TRG
CH0
CH1
INJ1_TRG
End of injected-1 conversion
End of injected-0 conversion
INJ0_TRG
CH0
CH1
INJ1_TRG
End of injected-1 conversion
▨ sampling
□ conversion

Injected conversion mode (INJ0 leads INJ1)


ADCx
End of injected-0 conversion
INJ0_TRG
CH0
CH1
INJ1_TRG
End of injected-1 conversion
End of injected-0 conversion
INJ0_TRG
CH0
CH1
INJ1_TRG
End of injected-1 conversion
▨ sampling
□ conversion

Injected conversion mode (INJ0 and INJ1 at the same time)


End of injected-0 conversion
INJ0_TRG
CH0
ADCx
CH1
INJ1_TRG
End of injected-1 conversion
End of injected-0 conversion
INJ0_TRG
CH0
CH1
INJ1_TRG
End of injected-1 conversion
sampling
conversion

Injected conversion mode (INJ1 leads INJ0)

8.9.4.7. Combined regular/injected simultaneous mode

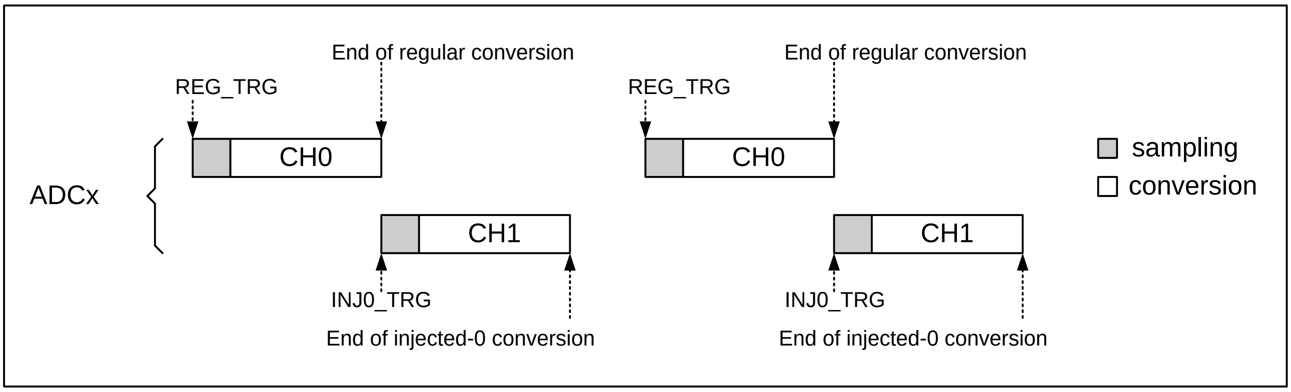
In combined regular and injected simultaneous mode, it is possible to interrupt regular channel simultaneous conversion to start injected trigger conversion of an injected channel. *Below Figures* show the behavior of an injected channel trigger interrupting a regular simultaneous conversion. The injected channel conversion is immediately started after the injected event triggers. If regular conversion is already running, it will be stopped. And regular channel resumed synchronously at the end of the injected conversion. Below explains how regular and injected mode works.

• If regular channel first converted:

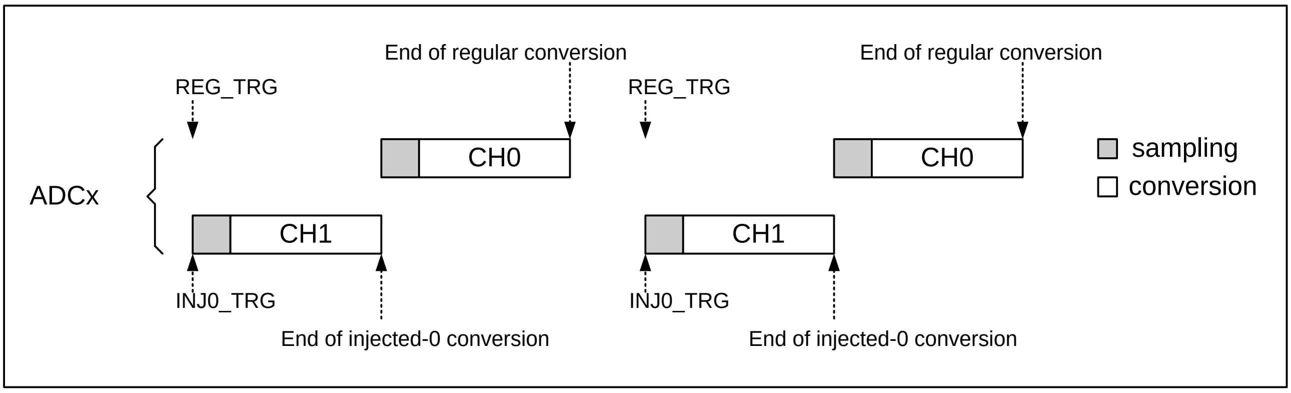
1. Regular channel is running.
2. If an injected trigger occurs during the regular channel conversion, the regular conversion is stopped and the injected channel is converted.
3. Then, the regular channel conversion is resumed at the end of the injected conversion.

• If injected channel first converted:

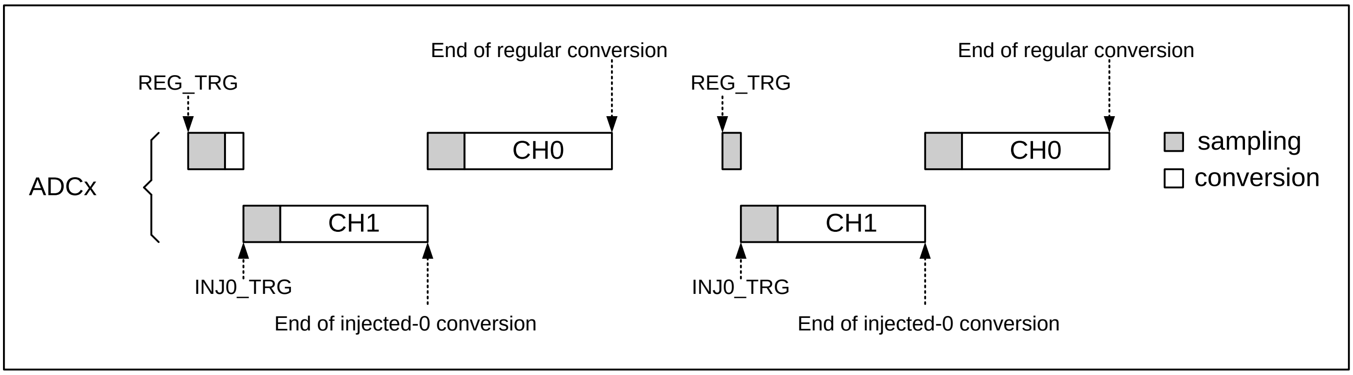
1. Injected channel is running.
2. If a regular trigger occurs during the injected channel conversion, the regular conversion is started at the end of the injected conversion.


End of regular conversion
REG_TRG
CH0
ADCx
CH1
INJ0_TRG
End of injected-0 conversion
End of regular conversion
REG_TRG
CH0
CH1
INJ0_TRG
End of injected-0 conversion
sampling
conversion

Combined regular/injected conversion mode (regular leads INJ0)


End of regular conversion
End of regular conversion
REG_TRG
REG_TRG
ADCx
CH0
CH0
CH1
CH1
sampling
conversion
INJ0_TRG
INJ0_TRG
End of injected-0 conversion
End of injected-0 conversion

Combined regular/injected conversion mode (regular and INJ0 at the same time)


End of regular conversion
End of regular conversion
REG_TRG
REG_TRG
ADCx
CH0
CH0
CH1
CH1
sampling
conversion
INJ0_TRG
INJ0_TRG
End of injected-0 conversion
End of injected-0 conversion

Combined regular/injected conversion mode (INJ0 interrupted regular)

8.9.4.8. Data alignment

In GPMQ9103A, ADC provides two data alignment type; user can set data alignment with left-aligned or right-aligned. **ADCx_CTRL0.DAT_ALIGN** bit selects the alignment of data stored after conversion. The following figures show some examples of the different data alignment.

D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0	0	0

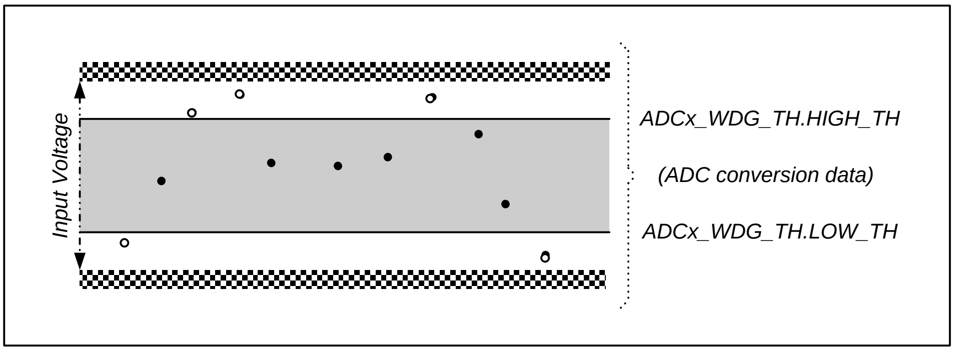
Left alignment of ADC data (**ADCx_CTRL0.DAT_ALIGN**=0)

0	0	0	0	0	0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Right alignment of ADC data (**ADCx_CTRL0.DAT_ALIGN**=1)

Analog watchdog

In GPMQ9103A, ADC provides a mechanism to measure voltage range of input signal. The **ADCx_CTRL0.A_WDGR_INTF/ADCx_CTRL0.A_WDGJ_INTF** are set if the analog voltage converted by the ADC is below **ADCx_WDG_TH.LOW_TH** or above **ADCx_WDG_TH.HIGH_TH**. Two interrupts can be enabled by using the **ADCx_CTRL0.A_WDGJ_INTE** and **ADCx_CTRL0.A_WDGR_INTE**. In addition, the threshold setting is independent of the **ADCx_WDG_TH** setting. The comparison is done before the data alignment.



Analog watchdog area

Register map

Base Address : 0x4012_0000				
Name	Description	Offset Address	Access	Reset value
ADC0_CTRL0	**ADC0 Control Register 0**	0x004	R/W	0x0000 0000
ADC0_CTRL1	**ADC0 Control Register 1**	0x008	R/W	0x0000 0000
ADC0_SMP0	**ADC0 Sample Time Control Register 0**	0x00C	R/W	0x0000 0000
ADC0_SMP1	**ADC0 Sample Time Control Register 1**	0x010	R/W	0x0000 0000
ADC0_REG_SEQ0	**ADC0 Regular Sequence Register 0**	0x014	R/W	0x0000 0000
ADC0_REG_SEQ1	**ADC0 Regular Sequence Register 1**	0x018	R/W	0x0000 0000
ADC0_INJ_SEQ	**ADC0 Injected Sequence Register**	0x01C	R/W	0x0000 0000
ADC0_WDG_TH	**ADC0 Watch-Dog Threshold Register**	0x020	R/W	0x0000 0000
ADC0_STS	**ADC0 Status Register**	0x024	R	0x0000 0001
ADC0_REG0_DAT	**ADC0 Regular 0 Data Register**	0x030	R	0x0000 0000
ADC0_REG1_DAT	**ADC0 Regular 1 Data Register**	0x034	R	0x0000 0000
ADC0_REG2_DAT	**ADC0 Regular 2 Data Register**	0x038	R	0x0000 0000
ADC0_REG3_DAT	**ADC0 Regular 3 Data Register**	0x03C	R	0x0000 0000
ADC0_REG4_DAT	**ADC0 Regular 4 Data Register**	0x040	R	0x0000 0000
ADC0_REG5_DAT	**ADC0 Regular 5 Data Register**	0x044	R	0x0000 0000
ADC0_REG6_DAT	**ADC0 Regular 6 Data Register**	0x048	R	0x0000 0000
ADC0_REG7_DAT	**ADC0 Regular 7 Data Register**	0x04C	R	0x0000 0000
ADC0_REG8_DAT	**ADC0 Regular 8 Data Register**	0x050	R	0x0000 0000
ADC0_REG9_DAT	**ADC0 Regular 9 Data Register**	0x054	R	0x0000 0000
ADC0_REG10_DAT	**ADC0 Regular 10 Data Register**	0x058	R	0x0000 0000
ADC0_REG11_DAT	**ADC0 Regular 11 Data Register**	0x05C	R	0x0000 0000
ADC0_INJ0_DAT	**ADC0 Injected0 Data Register**	0x070	R	0x0000 0000
ADC0_INJ1_DAT	**ADC0 Injected1 Data Register**	0x074	R	0x0000 0000
ADC0_INJ2_DAT	**ADC0 Injected2 Data Register**	0x078	R	0x0000 0000
ADC0_INJ3_DAT	**ADC0 Injected3 Data Register**	0x07C	R	0x0000 0000

Registers

ADCx Control Register0

ADC0_CTRL0 **ADC0 Control Register0** **Address : 0x4012 0004**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
SEQ_INTF	**A_WDGJ_INTF**	**A_WDGR_INTF**	**INJ3_INTF**	**INJ2_INTF**	**INJ1_INTF**	**INJ0_INTF**	**REG_INTF**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SEQ_INTE	**A_WDGJ_INTE**	**A_WDGR_INTE**	**INJ3_INTE**	**INJ2_INTE**	**INJ1_INTE**	**INJ0_INTE**	**REG_INTE**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--			**A_WDGJ_E**	**A_WDGR_E**	**DAT_ALIGN**	**--**	**LOOP_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--						**ADC_EN**	**SFT_STR**

Bit	Name	Description	Access	Reset value
[31]	**SEQ_INTF**	**Regular Sequence conversion completed interrupt flag** Read : 0 = no injection conversion is finished 1 = regular sequence conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x1
[30]	**A_WDGJ_INTF**	**Analog Watch-Dog interrupt flag on Injected channel** Read : 0 = no analog watch dog occurred 1 = analog watch dog occurred Write : 0 = no effect 1 = clear this flag	R/W	0x0
[29]	**A_WDGR_INTF**	**Analog Watch-Dog interrupt flag on regular channel** Read : 0 = no analog watch dog occurred 1 = analog watch dog occurred Write : 0 = no effect 1 = clear this flag	R/W	0x0
[28]	**INJ0_INTF**	**Injected conversion interrupt flag of channel 3** Read : 0 = no injection conversion is finished 1 = injection conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[27]	**INJ2_INTF**	**Injected conversion interrupt flag of channel 2** Read : 0 = no injection conversion is finished 1 = injection conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[26]	**INJ1_INTF**	**Injected conversion interrupt flag of channel 1**	R/W	0x0

Bit	Name	Description	Access	Reset value
		Read : 0 = no injection conversion is finished 1 = injection conversion has finished Write : 0 = no effect 1 = clear this flag		
[25]	**INJ0_INTF**	**Injected conversion interrupt flag of channel 0** Read : 0 = no injection conversion is finished 1 = injection conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[24]	**REG_INTF**	**Regular conversion interrupt flag** Read : 0 = no regular conversion is finished 1 = regular conversion has finished Write : 0 = no effect 1 = clear this flag	R/W	0x0
[23]	**--**	**Reserved**	R	0x0
[22]	**A_WDGJ_INTE**	**Analog Watch-Dog interrupt enable on Injected channel** 0 = disabled 1 = enabled	R/W	0x0
[21]	**A_WDGR_INTE**	**Analog Watch-Dog interrupt enable on regular channel** 0 = disabled 1 = enabled	R/W	0x0
[20]	**INJ3_INTE**	**Injected conversion interrupt enable of channel 3** 0 = disabled 1 = enabled	R/W	0x0
[19]	**INJ2_INTE**	**Injected conversion interrupt enable of channel 2** 0 = disabled 1 = enabled	R/W	0x0
[18]	**INJ1_INTE**	**Injected conversion interrupt enable of channel 1** 0 = disabled 1 = enabled	R/W	0x0
[17]	**INJ0_INTE**	**Injected conversion interrupt enable of channel 0** 0 = disabled 1 = enabled	R/W	0x0
[16]	**REG_INTE**	**Regular mode interrupt enable** 0 = disabled 1 = enabled	R/W	0x0
[15:13]	**--**	**Reserved**	R	0x0
[12]	**A_WDGJ_E**	**Analog Watchdog enable on Injected channel** 0 = disabled 1 = enabled	R/W	0x0
[11]	**A_WDGR_E**	**Analog Watchdog enable on regular channel** 0 = disabled	R/W	0x0

Bit	Name	Description	Access	Reset value
		1 = enabled		
[10]	**DAT_ALIGN**	**ADC output data alignment** 0 = left alignment 1 = right alignment	R/W	0x0
[9]	**--**	**Reserved**	R	0x0
[8]	**LOOP_E**	**ADC loop scan enable bit** 0 = single conversion mode 1 = loop scan conversion mode **Note:** This bit is available in regular mode only	R/W	0x0
[7:2]	**--**	**Reserved**	R	0x0
[1]	**ADC_E**	**ADC analog block enable** 0 = disabled 1 = enabled **Note:** User should set this bit enable before running ADC function.	R/W	0x0
[0]	**SFT_STR**	**Software start bit** 0 = disabled 1 = enabled (start conversion of regular channel) **Note:** This bit is available in regular mode only. This bit will be cleared by hardware, when LOOP_E is set to 0 and all of regular scan are be finished. If LOOP_E is set to 1, then signal be cleared by software.	R/W	0x0

ADCx Control Register1

ADC0_CTRL1 **ADC0 Control Register1** **Address : 0x4012 0008**

31	30	29	28	27	26	25	24
--		INJ3_TRG_SEL		--		INJ2_TRG_SEL	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--		INJ1_TRG_SEL		--		INJ0_TRG_SEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--				INJ3_E	INJ2_E	INJ1_E	INJ0_E
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
REG_CH_NUM				REG_SEQ_GAP			REG_E

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[31:30]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[29:28]</td><td><b>INJ3_TRG_SEL</b></td><td><b>Injection channel3 trigger source select bits</b>
<table>
<tr><th>INJ3_TRG_SEL[1:0]</th><th>Injection channel3 trigger source</th></tr>
<tr><td>00</td><td>Timer0 interrupt</td></tr>
<tr><td>01</td><td>Timer1 interrupt</td></tr>
<tr><td>10</td><td>Timer2 interrupt</td></tr>
<tr><td>11</td><td>Timer3 interrupt</td></tr>
</table>
<b>Note:</b> For INJy_TRG_SEL(y=0~3) are set to 3’b111. ADC0 is mapped to external interrupt 0, ADC1 is mapped to external interrupt 1.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[27:26]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[25:24]</td><td><b>INJ2_TRG_SEL</b></td><td><b>Injection channel2 trigger source select bits</b>
<table>
<tr><th>INJ2_TRG_SEL[1:0]</th><th>Injection channel2 trigger source</th></tr>
<tr><td>00</td><td>Timer0 interrupt</td></tr>
<tr><td>01</td><td>Timer1 interrupt</td></tr>
<tr><td>10</td><td>Timer2 interrupt</td></tr>
<tr><td>11</td><td>Timer3 interrupt</td></tr>
</table>
<b>Note:</b> For INJy_TRG_SEL(y=0~3) are set to 3’b111. ADC0 is mapped to external interrupt 0, ADC1 is mapped to external interrupt 1.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[23:22]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[21:20]</td><td><b>INJ1_TRG_SEL</b></td><td><b>Injection channel1 trigger source select bits</b>
<table>
<tr><th>INJ1_TRG_SEL[1:0]</th><th>Injection channel1 trigger source</th></tr>
<tr><td>00</td><td>Timer0 interrupt</td></tr>
<tr><td>01</td><td>Timer1 interrupt</td></tr>
<tr><td>10</td><td>Timer2 interrupt</td></tr>
<tr><td>11</td><td>Timer3 interrupt</td></tr>
</table>
<b>Note:</b> For INJy_TRG_SEL(y=0~3) are set to 3’b111. ADC0 is mapped to external interrupt 0, ADC1 is mapped to external interrupt 1.</td><td></td><td></td></tr>
<tr><td>[19:18]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[17:16]</td><td><b>INJ0_TRG_SEL</b></td><td><b>Injection channel0 trigger source select bits</b>
<table>
<tr><th>INJ0_TRG_SEL[1:0]</th><th>Injection channel0 trigger source</th></tr>
<tr><td>00</td><td>Timer0 interrupt</td></tr>
<tr><td>01</td><td>Timer1 interrupt</td></tr>
<tr><td>10</td><td>Timer2 interrupt</td></tr>
<tr><td>11</td><td>Timer3 interrupt</td></tr>
</table>
<b>Note:</b> For INJy_TRG_SEL(y=0~3) are set to 3’b111. ADC0 is mapped to</td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td></td><td></td><td>external interrupt 0, ADC1 is mapped to external interrupt 1.</td><td></td><td></td></tr>
<tr><td>[15:12]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[11]</td><td><b>INJ3_E</b></td><td><b>Injection channel 3 enable bit</b>
0 = disabled
1 = enable
<b>Note:</b> Before using injection sequence, user must set <b>ADCx_CTRL1.INJ3_E</b> to 1 first. Then set ADCx wait hardware trigger to start injection sequence.</td><td>R/W</td><td>0x00</td></tr>
<tr><td>[10]</td><td><b>INJ2_E</b></td><td><b>Injection channel 2 enable bit</b>
0 = disabled
1 = enabled
<b>Note:</b> Before use injection sequence, user must set <b>ADCx_CTRL1.INJ2_E</b> to 1 first. Then set ADCx wait hardware trigger to start injection sequence.</td><td>R/W</td><td>0x00</td></tr>
<tr><td>[9]</td><td><b>INJ1_E</b></td><td><b>Injection channel 1 enable bit</b>
0 = disabled
1 = enabled
<b>Note:</b> Before using injection sequence, user must set <b>ADCx_CTRL1.INJ1_E</b> to 1 first. Then set ADCx wait hardware trigger to start injection sequence.</td><td>R/W</td><td>0x00</td></tr>
<tr><td>[8]</td><td><b>INJ0_E</b></td><td><b>Injection channel 0 enable bit</b>
0 = disabled
1 = enabled
<b>Note:</b> Before using injection sequence, user must set <b>ADCx_CTRL1.INJ0_E</b> to 1 first. Then set ADCx wait hardware trigger to start injection sequence.</td><td>R/W</td><td>0x00</td></tr>
<tr><td>[7:4]</td><td><b>REG_CH_NUM</b></td><td><b>Channel number during regular sequence</b>
<b>Note:</b> The physical channel number is ADCx_CTRL1.REG_CH_NUM + 1. In addition, the scan sequence is always being started from sequence-0.
<b>Note:</b> The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.</td><td>R/W</td><td>0x0</td></tr>
<tr><td>[3:1]</td><td><b>REG_SEQ_GAP</b></td><td><b>Regular sequence gap select bits</b>
<table>
<tr><th>REG_SEQ_GAP[2:0]</th><th>Cycles of ADCx clock</th></tr>
<tr><td>000</td><td>0</td></tr>
<tr><td>001</td><td>1</td></tr>
<tr><td>010</td><td>2</td></tr>
<tr><td>011</td><td>3</td></tr>
<tr><td>100</td><td>4</td></tr>
<tr><td>101</td><td>5</td></tr>
<tr><td>110</td><td>6</td></tr>
<tr><td>111</td><td>7</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[0]</td><td><b>REG_E</b></td><td><b>Regular mode enable bit</b>
0 = disabled
1 = enabled
<b>Note:</b> Before using regular sequence, user must set <b>ADCx_CTRL1.REG_E</b> to 1 first. Then set <b>ADCx_CTRL0.SFT_STR</b> to 1 to start regular sequence.</td><td>R/W</td><td>0x0</td></tr>
</table>

ADCx Sample Time Control Register 0

ADC0_SMP0 **ADC0 Sample Time Control Register 0** **Address : 0x4012 000C**

31	30	29	28	27	26	25	24
CH7_SMP				CH6_SMP			
23	22	21	20	19	18	17	16
CH5_SMP				CH4_SMP			
15	14	13	12	11	10	9	8
CH3_SMP				CH2_SMP			
7	6	5	4	3	2	1	0
CH1_SMP				CH0_SMP			

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[31:28]</td><td><b>CH7_SMP</b></td><td><b>Channel7 sample time selection</b>
<table>
<tr><td>CH7_SMP[3:0]</td><td>Cycles of ADCx clock</td></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[27:24]</td><td><b>CH6_SMP</b></td><td><b>Channel6 sample time selection</b>
<table>
<tr><td>CH6_SMP[3:0]</td><td>Cycles of ADCx clock</td></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[23:20]</td><td><b>CH5_SMP</b></td><td><b>Channel5 sample time selection</b>
<table>
<tr><th>CH5_SMP[3:0]</th><th>Cycles of ADCx clock</th></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[19:16]</td><td><b>CH4_SMP</b></td><td><b>Channel4 sample time selection</b>
<table>
<tr><th>CH4_SMP[3:0]</th><th>Cycles of ADCx clock</th></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[15:12]</td><td><b>CH3_SMP</b></td><td><b>Channel3 sample time selection</b>
<table>
<tr><th>CH3_SMP[3:0]</th><th>Cycles of ADCx clock</th></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td></td><td></td><td><table>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td></td><td></td></tr>
<tr><td>[11:8]</td><td><b>CH2_SMP</b></td><td><b>Channel2 sample time selection</b><table>
<tr><td>CH2_SMP[3:0]</td><td>Cycles of ADCx clock</td></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[7:4]</td><td><b>CH1_SMP</b></td><td><b>Channel1 sample time selection</b><table>
<tr><td>CH1_SMP[3:0]</td><td>Cycles of ADCx clock</td></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[3:0]</td><td><b>CH0_SMP</b></td><td><b>Channel0 sample time selection</b><table>
<tr><td>CH0_SMP[3:0]</td><td>Cycles of ADCx clock</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td></td><td></td><td><table>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td></td><td></td></tr>
</table>

ADCx Sample Time Control Register 1

ADC0_SMP1	ADC0 Sample Time Control Register 1						Address : 0x4012 0010
31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CH11_SMP				**CH10_SMP**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH9_SMP				**CH8_SMP**			

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td>[31:16]</td><td>--</td><td><b>Reserved</b></td><td>R</td><td>0x0</td></tr>
<tr><td>[15:12]</td><td><b>CH11_SMP</b></td><td><b>Channel11 sample time selection</b>
<table>
<tr><th>CH11_SMP[3:0]</th><th>Cycles of ADC clock</th></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
<tr><td>1110</td><td>100</td></tr>
<tr><td>1111</td><td>200</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[11:8]</td><td><b>CH10_SMP</b></td><td><b>Channel10 sample time selection</b>
<table>
<tr><th>CH10_SMP[3:0]</th><th>Cycles of ADC clock</th></tr>
<tr><td>0000</td><td>1</td></tr>
<tr><td>0001</td><td>2</td></tr>
<tr><td>0010</td><td>3</td></tr>
<tr><td>0011</td><td>4</td></tr>
<tr><td>0100</td><td>5</td></tr>
<tr><td>0101</td><td>6</td></tr>
<tr><td>0110</td><td>7</td></tr>
<tr><td>0111</td><td>8</td></tr>
<tr><td>1000</td><td>9</td></tr>
<tr><td>1001</td><td>10</td></tr>
<tr><td>1010</td><td>15</td></tr>
<tr><td>1011</td><td>20</td></tr>
<tr><td>1100</td><td>30</td></tr>
<tr><td>1101</td><td>50</td></tr>
</table></td><td>R/W</td><td>0x0</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th><th>Access</th><th>Reset value</th></tr>
<tr><td></td><td></td><td><table><tr><td>1110</td><td>100</td></tr><tr><td>1111</td><td>200</td></tr></table></td><td></td><td></td></tr>
<tr><td>[7:4]</td><td><b>CH9_SMP</b></td><td><b>Channel9 sample time selection</b><table><tr><td>CH9_SMP[3:0]</td><td>Cycles of ADC clock</td></tr><tr><td>0000</td><td>1</td></tr><tr><td>0001</td><td>2</td></tr><tr><td>0010</td><td>3</td></tr><tr><td>0011</td><td>4</td></tr><tr><td>0100</td><td>5</td></tr><tr><td>0101</td><td>6</td></tr><tr><td>0110</td><td>7</td></tr><tr><td>0111</td><td>8</td></tr><tr><td>1000</td><td>9</td></tr><tr><td>1001</td><td>10</td></tr><tr><td>1010</td><td>15</td></tr><tr><td>1011</td><td>20</td></tr><tr><td>1100</td><td>30</td></tr><tr><td>1101</td><td>50</td></tr><tr><td>1110</td><td>100</td></tr><tr><td>1111</td><td>200</td></tr></table></td><td>R/W</td><td>0x0</td></tr>
<tr><td>[3:0]</td><td><b>CH8_SMP</b></td><td><b>Channel8 sample time selection</b><table><tr><td>CH8_SMP[3:0]</td><td>Cycles of ADC clock</td></tr><tr><td>0000</td><td>1</td></tr><tr><td>0001</td><td>2</td></tr><tr><td>0010</td><td>3</td></tr><tr><td>0011</td><td>4</td></tr><tr><td>0100</td><td>5</td></tr><tr><td>0101</td><td>6</td></tr><tr><td>0110</td><td>7</td></tr><tr><td>0111</td><td>8</td></tr><tr><td>1000</td><td>9</td></tr><tr><td>1001</td><td>10</td></tr><tr><td>1010</td><td>15</td></tr><tr><td>1011</td><td>20</td></tr><tr><td>1100</td><td>30</td></tr><tr><td>1101</td><td>50</td></tr><tr><td>1110</td><td>100</td></tr><tr><td>1111</td><td>200</td></tr></table></td><td>R/W</td><td>0x0</td></tr>
</table>

ADCx Regular Sequence Register0

ADC0_REG_SEQ0 **ADC0 Regular Sequence Register0** **Address : 0x4012 0014**

31	30	29	28	27	26	25	24
REG_SEQ7				REG_SEQ6			
23	22	21	20	19	18	17	16
REG_SEQ5				REG_SEQ4			
15	14	13	12	11	10	9	8
REG_SEQ3				REG_SEQ2			
7	6	5	4	3	2	1	0
REG_SEQ1				REG_SEQ0			

Bit	Name	Description	Access	Reset value
[31:28]	**REG_SEQ7**	**The 7th conversion in regular sequence** This indicates which channel in 7th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[27:24]	**REG_SEQ6**	**The 6th conversion in regular sequence** This indicates which channel in 6th regular sequence **Note:** The maximum value is the channel number. When written value exceed 0xb, this filed will not be updated.	R/W	0x0
[23:20]	**REG_SEQ5**	**The 5th conversion in regular sequence** This indicates which channel in 5th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[19:16]	**REG_SEQ4**	**The 4th conversion in regular sequence** This indicates which channel in 4th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[15:12]	**REG_SEQ3**	**The 3rd conversion in regular sequence** This indicates which channel in 3rd regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[11:8]	**REG_SEQ2**	**The 2nd conversion in regular sequence** This indicates which channel in 2nd regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[7:4]	**REG_SEQ1**	**The 1st conversion in regular sequence** This indicates which channel in 1st regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[3:0]	**REG_SEQ0**	**The 0th conversion in regular sequence** This indicates which channel in 0th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0

ADCx Regular Sequence Register1

ADC0_REG_SEQ1 ADC0 Regular Sequence Register1 Address : 0x4012 0018

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
REG_SEQ11				REG_SEQ10			
7	6	5	4	3	2	1	0
REG_SEQ9				REG_SEQ8			

Bit	Name	Description	Access	Reset value
[31:16]	**Reserved**	R	0x0	**Reserved**
[15:12]	**REG_SEQ11**	**The 11th conversion in regular sequence** This indicates which channel in 11th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[11:8]	**REG_SEQ10**	**The 10th conversion in regular sequence** This indicates which channel in 10th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[7:4]	**REG_SEQ9**	**The 9th conversion in regular sequence** This indicates which channel in 9th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0
[3:0]	**REG_SEQ8**	**The 8th conversion in regular sequence** This indicates which channel in 8th regular sequence **Note:** The maximum value is the channel number. When the written value exceeds 0xb, this filed will not be updated.	R/W	0x0

ADCx Injected Sequence Register

ADC0_INJ_SEQ **ADC0 Injected Sequence Register** **Address : 0x4012 001C**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INJ_SEQ3				INJ_SEQ2			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INJ_SEQ1				INJ_SEQ0			

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R	0x0000
[15:12]	**INJ_SEQ3**	**The 3rd conversion in injected sequence** This indicates which channel in 3rd injected sequence	R/W	0x0
[11:8]	**INJ_SEQ2**	**The 2nd conversion in injected sequence** This indicates which channel in 2nd injected sequence	R/W	0x0
[7:4]	**INJ_SEQ1**	**The 1st conversion in injected sequence** This indicates which channel in 1st injected sequence	R/W	0x0
[3:0]	**INJ_SEQ0**	**The 0th conversion in injected sequence** This indicates which channel in 0th injected sequence	R/W	0x0

ADCx Watch-Dog Threshold Register

ADC0_WDG_TH **ADC0 Watch-Dog Threshold Register** **Address : 0x4012 0020**

31	30	29	28	27	26	25	24
						HIGH_TH[9:8]	
23	22	21	20	19	18	17	16
HIGH_TH[7:0]							
15	14	13	12	11	10	9	8
						LOW_TH[9:8]	
7	6	5	4	3	2	1	0
LOW_TH[7:0]							

Bit	Name	Description	Access	Reset value
[31:26]	--	Reserved	R	0x0
[25:16]	HIGH_TH	Analog watch-dog high threshold	R/W	0x000
[15:10]	--	Reserved	R	0x0
[9:0]	LOW_TH	Analog watch-dog low threshold	R/W	0x000

ADC0 Status Register

ADC0_STS **ADC0 Status Register** **Address : 0x4012 0024**

31	30	29	28	27	26	25	24
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
							ADC_RDY

Bit	Name	Description	Access	Reset value
[31:1]	--	**Reserved**	R	0x0
[0]	**ADC_RDY**	**ADC Data Ready Signal** 0 = ADC data not ready. 1 = ADC data ready.	R	0x1

ADC0 Regular Data Register

ADC0_REG0_DAT	ADC0 Regular 0 Data Register	Address : 0x4012 0030
ADC0_REG1_DAT	ADC0 Regular 1 Data Register	Address : 0x4012 0034
ADC0_REG2_DAT	ADC0 Regular 2 Data Register	Address : 0x4012 0038
ADC0_REG3_DAT	ADC0 Regular 3 Data Register	Address : 0x4012 003C
ADC0_REG4_DAT	ADC0 Regular 4 Data Register	Address : 0x4012 0040
ADC0_REG5_DAT	ADC0 Regular 5 Data Register	Address : 0x4012 0044
ADC0_REG6_DAT	ADC0 Regular 6 Data Register	Address : 0x4012 0048
ADC0_REG7_DAT	ADC0 Regular 7 Data Register	Address : 0x4012 004C
ADC0_REG8_DAT	ADC0 Regular 8 Data Register	Address : 0x4012 0050
ADC0_REG9_DAT	ADC0 Regular 9 Data Register	Address : 0x4012 0054
ADC0_REG10_DAT	ADC0 Regular 10 Data Register	Address : 0x4012 0058
ADC0_REG11_DAT	ADC0 Regular 11 Data Register	Address : 0x4012 005C

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADC0_REG_x_ DATA[15:8]							
7	6	5	4	3	2	1	0
ADC0_REG_x_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	ADC0_REG_x_DATA	ADC0 regular X data Note: X = 0 ~ 11.	R	0x0000

:

ADCx Injected0 Data Register

ADC0_INJ0_DAT **ADC0 Injected0 Data Register** **Address : 0x4012 0070**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADCx_INJ0_DATA[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADCx_INJ0_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R	0x0000
[15:0]	**ADC0_INJx_DATA**	**ADCx injected0 data**	R	0x0000

ADCx Injected1 Data Register

ADC0_INJ1_DAT ADC0 Injected1 Data Register Address : 0x4012 0074

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADCx_INJ1_DATA[15:8]							
7	6	5	4	3	2	1	0
ADCx_INJ1_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	Reserved	R	0x0000
[15:0]	ADCx_INJ1_DATA	ADCx injected1 data	R	0x0000

ADCx Injected2 Data Register

ADC0_INJ2_DAT ADC0 Injected2 Data Register Address : 0x4012 0078

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADCx_INJ2_DATA[15:8]							
7	6	5	4	3	2	1	0
ADCx_INJ2_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R	0x0000
[15:0]	**ADCx_INJ2_DATA**	**ADCx injected2 data**	R	0x0000

ADCx Injected3 Data Register

ADC0_INJ3_DAT **ADC0 Injected3 Data Register** **Address : 0x4012 007C**

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
ADCx_INJ3_DATA[15:8]							
7	6	5	4	3	2	1	0
ADCx_INJ3_DATA[7:0]							

Bit	Name	Description	Access	Reset value
[31:16]	--	**Reserved**	R	0x0000
[15:0]	**ADCx_INJ3_DATA**	**ADCx injected3 data**	R	0x0000

8.10. Inter integrated circuit interface (I2C)

8.10.1. Overview

The GPMQ9103A incorporates One I2C interface. The inter integrated circuit interface (I^2C) is used for attaching lower-speed peripheral ICs to processors and microcontrollers in short-distance.

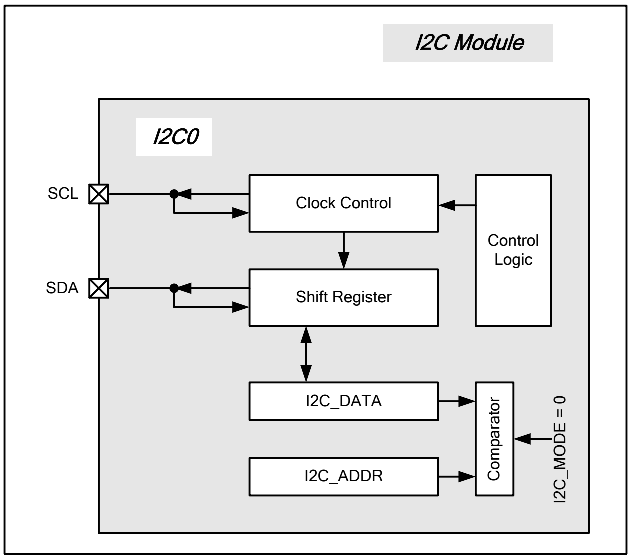
The I2C uses only two bidirectional open-drain lines, that are serial data line (SDA) and serial clock line (SCL). Logic 0 is output by sinking the bus to ground, and logic 1 is output by letting the bus to floating state, and via the pull-up resistor pulls it to high.

8.10.2. Features

- Multi-master capability
- Supports I2C master and slave mode
- Optional clock

8.10.3. Block Diagram

The figure is the block diagram of I2C.



I2C block diagram

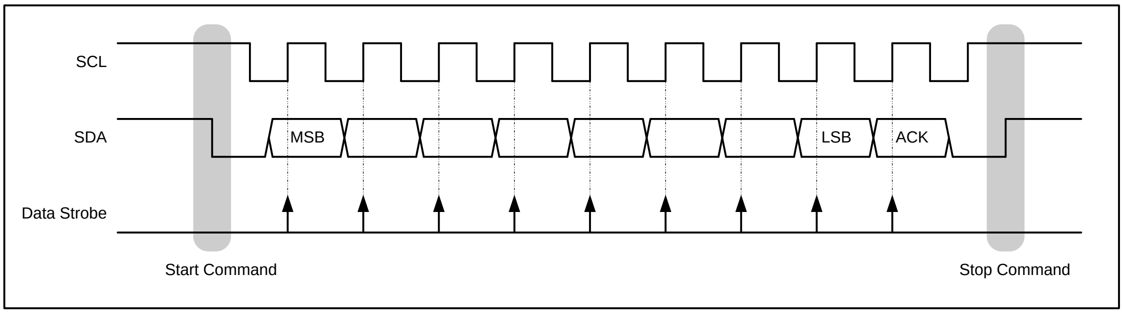
Function Description

In GPMQ9103A, I2C supports 7-bit or 10-bit (depending on the device used) address space. Only two wires (SCK and SDA) are needed to implement the protocol. In multi-master I2C-bus mode, multiple microprocessors can receive or transmit serial data to or from slave devices. If more than one master simultaneously tries to control the line, an arbitration mechanism is used to judge which one is bus owner. In I2C controller, four transfer modes are supported:

- master transmit – master is sending data to a slave.
- master receive – master is receiving data from a slave.
- slave transmit – slave is sending data to the master.
- slave receive – slave is receiving data from the master.

Formats of I2C frame

Figure shows the frame format of I2C.

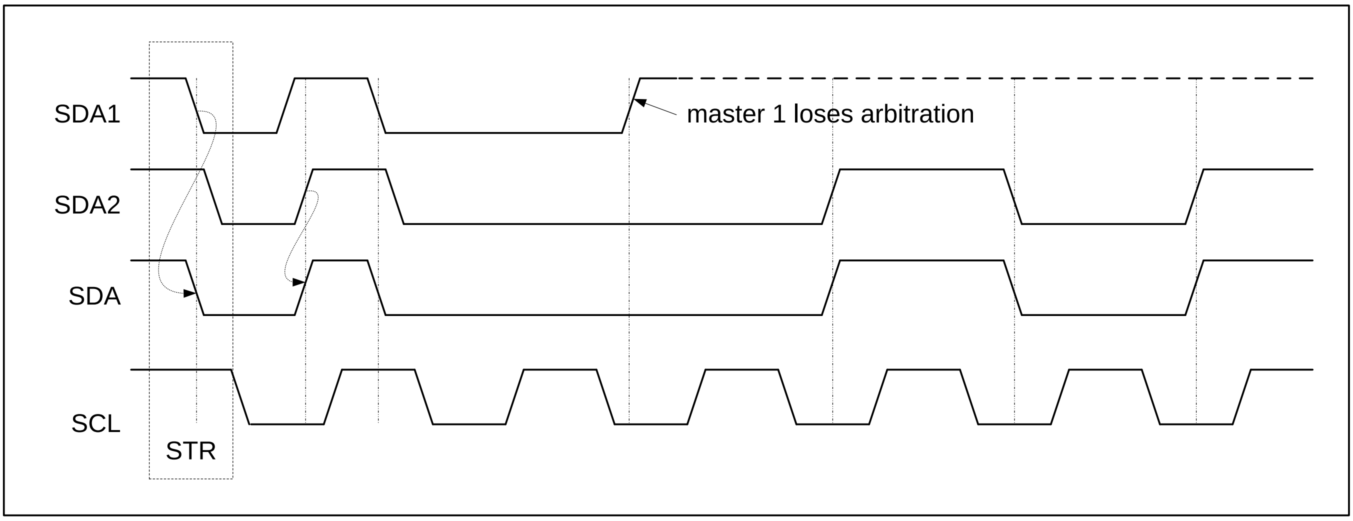


The frame format of I2C

8.10.4. Master mode

8.10.4.1. Arbitration lost

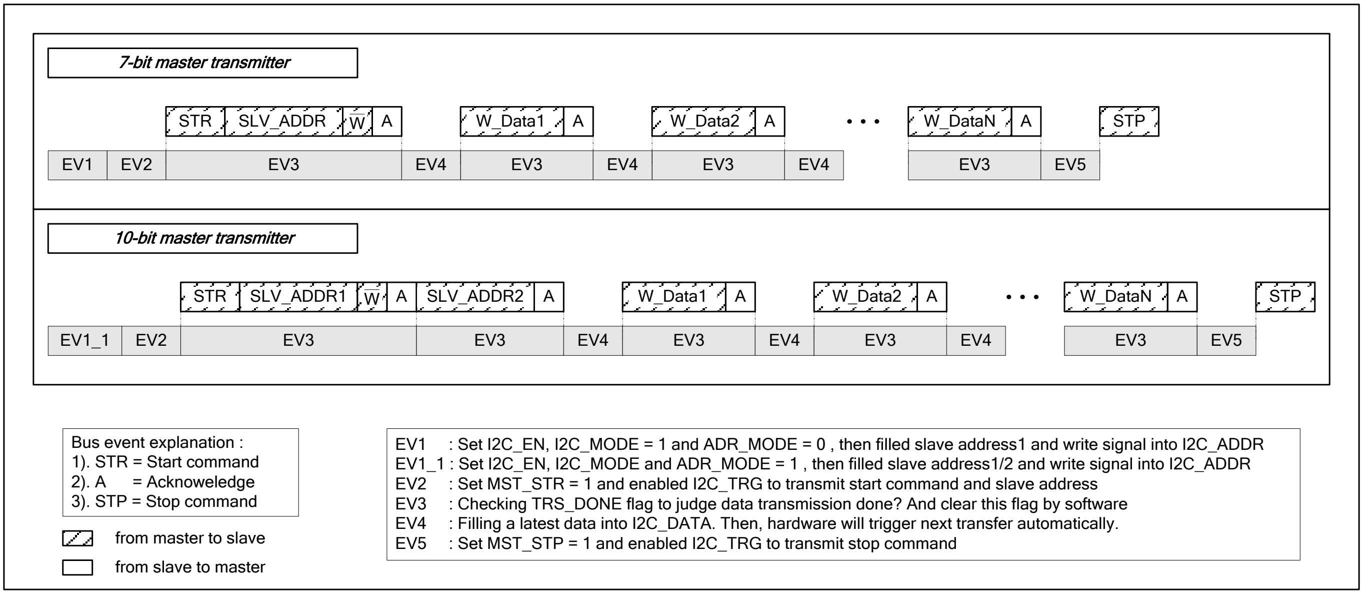
A master may start a transfer only if the bus is free. Arbitration takes place on the SDA line, while the SCL line is at the HIGH level; while another master is transmitting a LOW level will switch off its DATA output stage. The **I2Cx_STS.ARB_LOST** is set by hardware when the I2C interface detects an arbitration lost. Then, I2C controller switches from master to slave mode automatically. A master that still generate clock pulses until the end of the packet when it loses the arbitration. Figure shows an arbitration mechanism of two masters.



Arbitration mechanism of two masters.

8.10.4.2. I2C master transmit with manual mode

In master mode, before using I2C transmits function, user must fill in transfer data and set **I2Cx_CTRL.I2C_EN** to 1. In addition, master can indicates slave which is a write transfer by set **I2Cx_ADDR.R_W** to 0. Then, I2C will send out data immediately when set **I2Cx_CTRL.I2C_TRG**. During an I2C transmission, data shifts out most significant bit first on the SDA pin. The **I2Cx_STS.TRS_DONE** flag will be set to 1 when acknowledge bits is finished. Before next transmission, user must clear this bit by CPU. Next, fill in latest data into **I2Cx_DATA** that will trigger next transfer. If user wants to terminate I2C transmission, set **I2Cx_CTRL.MST_STP** and **I2Cx_CTRL.I2C_TRG** to 1 that will induce a stop command transfer. Figure shows a master transmit with manual mode.



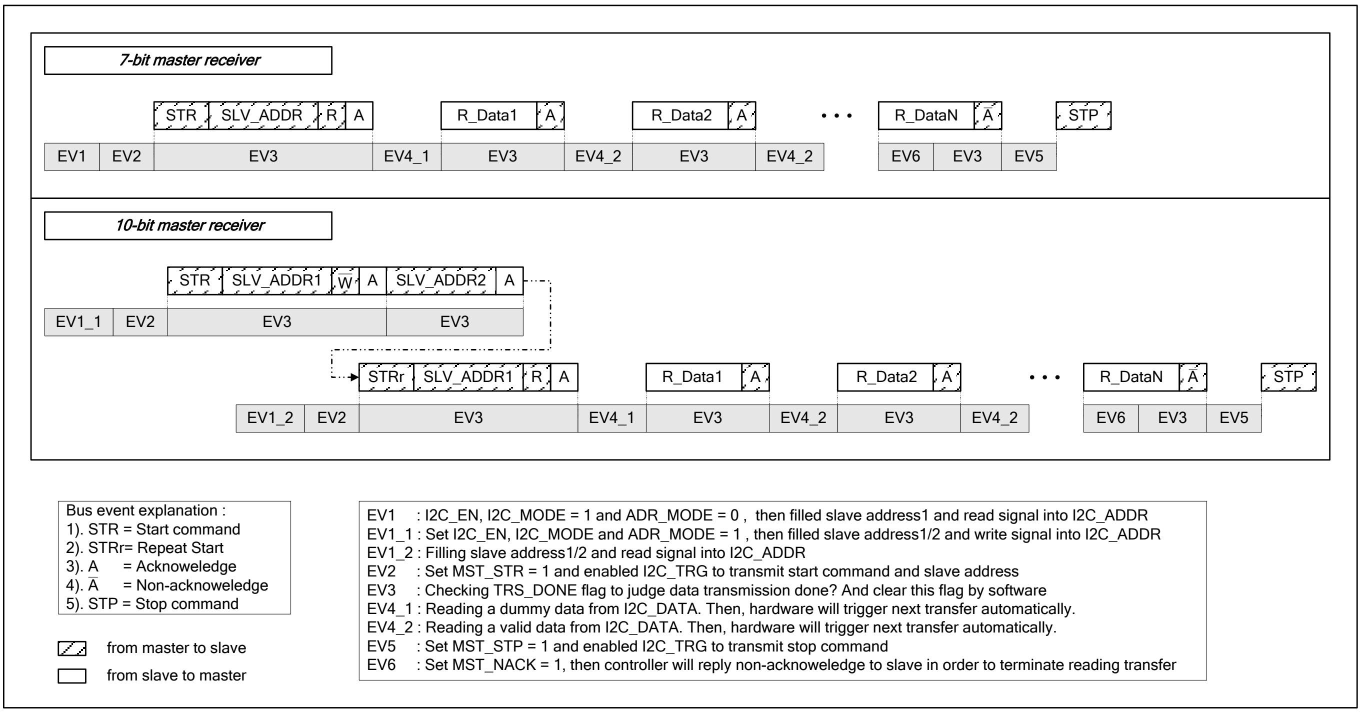
Bus event explanation :
1). STR = Start command
2). A = Acknoweledge
3). STP = Stop command

EV1 : Set I2C_EN, I2C_MODE = 1 and ADR_MODE = 0 , then filled slave address1 and write signal into I2C_ADDR
EV1_1 : Set I2C_EN, I2C_MODE and ADR_MODE = 1 , then filled slave address1/2 and write signal into I2C_ADDR
EV2 : Set MST_STR = 1 and enabled I2C_TRG to transmit start command and slave address
EV3 : Checking TRS_DONE flag to judge data transmission done? And clear this flag by software
EV4 : Filling a latest data into I2C_DATA. Then, hardware will trigger next transfer automatically.
EV5 : Set MST_STP = 1 and enabled I2C_TRG to transmit stop command

I2C master transmit with manual mode

8.10.4.3. I2C master receive with manual mode

In master mode, before using I2C receives function, user must fill in transfer data and set **I2Cx_CTRL.I2C_EN** to 1. In addition, master can indicates slave which is a read transfer by set **I2Cx_ADDR.R_W** to 1. Then, I2C will send out clock when set **I2Cx_CTRL.I2C_TRG**. During an I2C transmission, data shifts in most significant bit first from the SDA pin. The **I2Cx_STS.TRS_DONE** will be set to 1 when acknowledge bits is finished. Before next transmission, user must clear this bit by CPU. Next, read data from **I2Cx_DATA** that will trigger next transfer. If user wants to terminate I2C transmission, set **I2Cx_CTRL.MST_NACK** to 1 during final read transfer. After final transfer done, set **I2Cx_CTRL.MST_STP** and **I2Cx_CTRL.I2C_TRG** to 1 that will induce a stop command transfer. Below figure shows a master receive with manual mode.



Bus event explanation :
1). STR = Start command
2). STRr= Repeat Start
3). A = Acknoweledge
4). $\overline{A}$ = Non-acknoweledge
5). STP = Stop command

▨ from master to slave
□ from slave to master

EV1 : I2C_EN, I2C_MODE = 1 and ADR_MODE = 0 , then filled slave address1 and read signal into I2C_ADDR
EV1_1 : Set I2C_EN, I2C_MODE and ADR_MODE = 1 , then filled slave address1/2 and write signal into I2C_ADDR
EV1_2 : Filling slave address1/2 and read signal into I2C_ADDR
EV2 : Set MST_STR = 1 and enabled I2C_TRG to transmit start command and slave address
EV3 : Checking TRS_DONE flag to judge data transmission done? And clear this flag by software
EV4_1 : Reading a dummy data from I2C_DATA. Then, hardware will trigger next transfer automatically.
EV4_2 : Reading a valid data from I2C_DATA. Then, hardware will trigger next transfer automatically.
EV5 : Set MST_STP = 1 and enabled I2C_TRG to transmit stop command
EV6 : Set MST_NACK = 1, then controller will reply non-acknoweledge to slave in order to terminate reading transfer

I2C master receive with manual mode

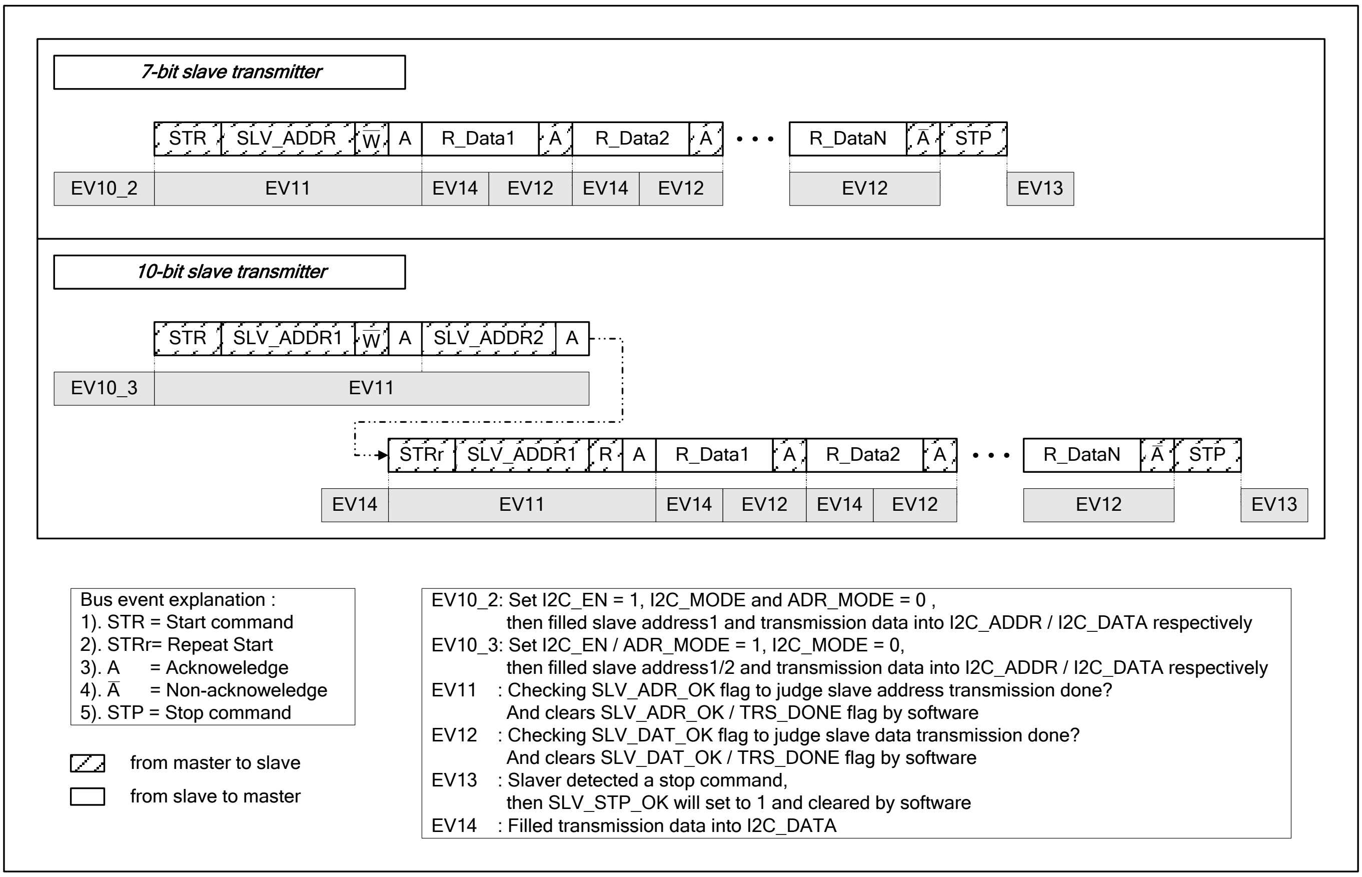
8.10.5. Slave mode

8.10.5.1. General call

In I2C bus, a ‘general call’ address which can address all devices. All devices should respond acknowledge when this address is used. However, devices can be made to ignore this address. The **I2Cx_STS.GEN_CALL** will be set when slave received a general call. CPU can receive an interrupt request if **I2Cx_CTRL.I2C_INTE** is set to 1.

8.10.5.2. I2C slave transmit with manual mode

In slave mode, before receiving data from master, user must fill in slave-address and set **I2Cx_CTRL.I2C_EN** to 1. Slave controller will set **I2Cx_STS.SLV_ADR_OK** flag to 1 if the received data of slave address is matched **I2Cx_ADDR**. Else **I2Cx_STS.I2C_ADR_ERR** flag will be set to 1. After slave-address checking, slave controller starts to receive input data that from the master. During an I2C transmission, the **I2Cx_STS.TRS_DONE** and **I2Cx_STS.SLV_DAT_OK** flag will be set to 1. User must clear these flags by CPU. Next, read the receiving data from **I2Cx_DATA**. The transfer will execute until a stop command is received. Figure 10-8 shows a slave transmit with manual mode.



Bus event explanation :
1). STR = Start command
2). STRr= Repeat Start
3). A = Acknoweledge
4). $\bar{A}$ = Non-acknoweledge
5). STP = Stop command

▨ from master to slave
□ from slave to master

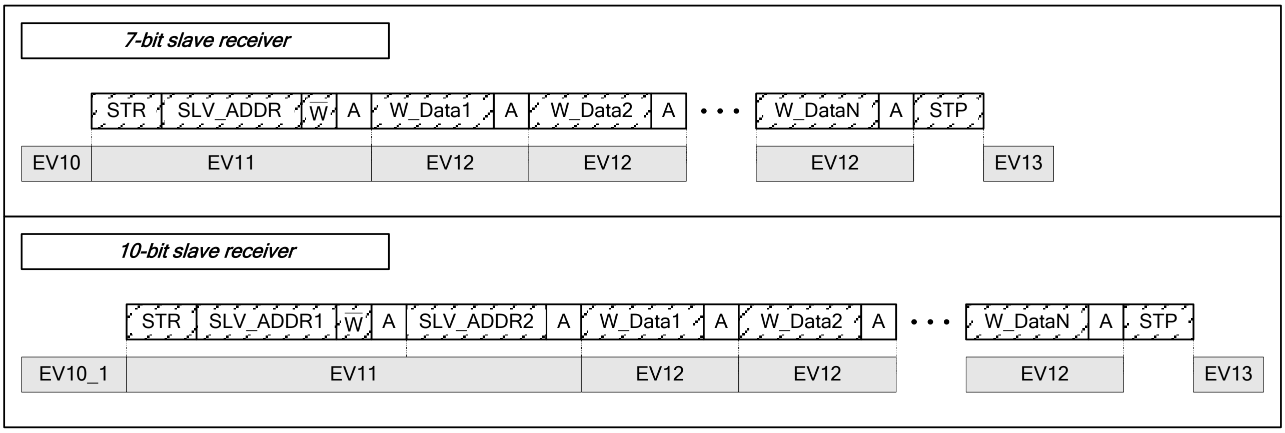
EV10_2: Set I2C_EN = 1, I2C_MODE and ADR_MODE = 0 ,
then filled slave address1 and transmission data into I2C_ADDR / I2C_DATA respectively
EV10_3: Set I2C_EN / ADR_MODE = 1, I2C_MODE = 0,
then filled slave address1/2 and transmission data into I2C_ADDR / I2C_DATA respectively
EV11 : Checking SLV_ADR_OK flag to judge slave address transmission done?
And clears SLV_ADR_OK / TRS_DONE flag by software
EV12 : Checking SLV_DAT_OK flag to judge slave data transmission done?
And clears SLV_DAT_OK / TRS_DONE flag by software
EV13 : Slaver detected a stop command,
then SLV_STP_OK will set to 1 and cleared by software
EV14 : Filled transmission data into I2C_DATA

I2C slave transmit with manual mode

8.10.5.3. I2C slave receive with manual mode

In slave mode, before using I2C receives function, user must fill in slave-address, transfer data and set **I2Cx_CTRL.I2C_EN** to 1. Slave controller will set **I2Cx_STS.SLV_ADR_OK** flag to 1 if the received data of slave address is matched **I2Cx_ADDR**. Else **I2Cx_STS.SLV_ADR_ERR** flag will be set to 1. After slave-address checking, filled-in data into **I2Cx_DATA** and waits **I2Cx_STS.SLV_DAT_OK** flag to be set to 1. The transfer will execute until a stop command is received.

Figure 10-9 shows a slave receive with manual mode.



Bus event explanation :
1). STR = Start command
2). A = Acknoweledge
3). STP = Stop command

▨ from master to slave

□ from slave to master

EV10 : Set I2C_EN = 1, I2C_MODE and ADR_MODE = 0 , then filled slave address1 into I2C_ADDR
EV10_1: Set I2C_EN / ADR_MODE = 1, I2C_MODE = 0,
then filled slave address1/2 into I2C_ADDR
EV11 : Checking SLV_ADR_OK flag to judge slave address transmission done?
And clears SLV_ADR_OK / TRS_DONE flag by software
EV12 : Checking SLV_DAT_OK flag to judge slave data transmission done?
And clears SLV_DAT_OK / TRS_DONE flag by software
EV13 : Slaver detected a stop command,
then SLV_STP_OK will set to 1 and cleared by sofwtare

I2C slave receive with manual mode

Register map

Base Address : 0x400B_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
I2C0_CTRL	I2C 0 Control Register	0x000	R/W	0x0000 0000
I2C0_STS	I2C 0 Status Register	0x004	R/W	0x0000 0002
I2C0_ADDR	I2C 0 Transmit Data Register	0x008	R/W	0x0000 00C2
I2C0_DATA	I2C 0 Receive Data Register	0x00C	R/W	0x0000 0000

Registers

I2C0 Control Register

I2C0_CTRL **I2C 0 Control Register** **Address : 0x400B 0000**

31	30	29	28	27	26	25	24
--							I2C_MODE
23	22	21	20	19	18	17	16
ERR_SADR_IE	I2C_INTE	I2C_DB_TIME					
15	14	13	12	11	10	9	8
MST_STR	MST_STP	MST_NACK	--		I2C_CLK_SEL		I2C_EN
7	6	5	4	3	2	1	0
--							I2C_TRG

Bit	Name	Description	Access	Reset value
[31:25]	--	**Reserved**	R	0x0
[24]	**I2C_MODE**	**I2C controller operating mode select bits** 0 = slaver mode 1 = master mode	R/W	0x0
[23]	**ERR_SADR_IE**	**Slaver address error interrupt enable bit** 0 = disabled 1 = enabled **Note :** This bit is available when I2C controller operating in slaver mode.	R/W	0x0
[22]	**I2C_INTE**	**I2C interrupt enable bit** 0 = disabled 1 = enabled	R/W	0x0
[21:16]	**I2C_DB_TIME**	**SCL / SDA input de-bounce time select bits**	R/W	0x0
[15]	**MST_STR**	**I2C controller issued start command enable bit** 0 = disabled 1 = enabled **Note :**This bit will be cleared automatically when this transfer is finished.	R/W	0x0
[14]	**MST_STP**	**I2C controller issued stop command enable bit** 0 = disabled 1 = enabled **Note :**This bit will be cleared automatically when this transfer is finished.	R/W	0x0
[13]	**MST_NACK**	**I2C controller issued non-acknowledge enable bit** 0 = disabled 1 = enabled **Note :**This bit will be cleared automatically when this transfer is finished.	R/W	0x0
[12:11]	--	**Reserved**	R	0x0
[10:9]	**I2C_CK_SEL**	**I2C controller serial clock select bits** I2C_CK_SEL[1:0] — Clock Source 00 — I2C clock is system clock / 128 01 — I2C clock is system clock / 256 10 — I2C clock is system clock / 768 11 — I2C clock is system clock / 1024	R/W	0x0
[8]	**I2C_EN**	**I2C controller enable bit** 0 = disabled 1 = enabled	R/W	0x0

Bit	Name	Description	Access	Reset value
[7:1]	**--**	**Reserved**	R	0x00
[0]	**I2C_TRG**	**The start transmission trigger bit** This bit is for master mode only. The I2C master will begin to transmit or receive data when I2C_EN is set to 1. This bit will be cleared by H/W automatically. 0 = disabled 1 = enabled **Note**: This bit is available when **I2C_CTRL.I2C_EN** is enabled.	R/W	0x0

I2C0 Status Register

I2C0_STS I2C 0 Status Register Address : 0x400B 0004

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--			SCL_9TH	SLV_ADR_ERR	BUSY	GEN_CALL	--
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--	ARB_LOST	--	SLV_ADR_OK	SLV_DAT_OK	SLV_STP_OK	NO_ACK	TRS_DONE

Bit	Name	Description	Access	Reset value
[31:12]	--	**Reserved**	R	0x0
[12]	**SCL_9TH**	**SCL 9th clock period** 0 = N/A 1 = SCL 9th clock period	R	0x0
[11]	**SLV_ADR_OK**	**Slaved address error flag** read: 0 = slaver address is correct address 1 = slaver address is wrong address write: 0 = no effect 1 = clear this bit	R/W	0x0
[10]	**BUSY**	**I2C controller busy flag** read: 0 = No communication on the bus 1 = Communication ongoing on the bus write: 0 = no effect 1 = clear this bit	R/W	0x0
[9]	**GEN_CALL**	**I2C general call flag** read: 0 = I2C master has issued general call 1 = I2C master has not issued general call write: 0 = no effect 1 = clear this bit	R/W	0x0
[8:7]	--	**Reserved**	R	0x0
[6]	**ARB_LOST**	**I2C bus arbitration lost flag** read: 0 = I2C bus arbitration lost is not occurred 1 = I2C bus arbitration lost is occurred write: 0 = no effect 1 = clear this bit **Note:** This error occurs when the I2C interface detects an arbitration lost condition.	R/W	0x0
[5]	--	**Reserved**	R	0x0

Bit	Name	Description	Access	Reset value
[4]	**SLV_ADR_OK**	**Slaved address had been received** read: 0 = slaver address is not asserted 1 = slaver address is asserted and match to the setting write: 0 = no effect 1 = clear this bit	R/W	0x0
[3]	**SLV_DAT_OK**	Data transmit or receive done flag read: 0 = data is transmitting or idle now 1 = data is transmission complete write: 0 = no effect 1 = clear this bit **Note:** This bit is used to slaver mode only.	R/W	0x0
[2]	**SLV_STP_OK**	**Stop command had received flag** read: 0 = stop command is not be received 1 = stop command had received write: 0 = no effect 1 = clear this bit **Note:** This bit is used to slaver mode only.	R/W	0x0
[1]	**NO_ACK**	I2C have not received acknowledging signal. read : 0 = acknowledge 1 = no acknowledge	R/W	0x1
[0]	**TRS_DONE**	**I2C controller transmission complete flag** read: 0 = I2C is idle or on going 1 = I2C finished data transmission write: 0 = no effect 1 = clear this bit	R/W	0x0

I2C0 Address Register

I2C0_ADDR **I2C 0 Address Register** **Address : 0x400B 0008**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							ADR_MODE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDR2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDR1							R_W

Bit	Name	Description	Access	Reset value
[31:17]	--	**Reserved**	R/W	0x0000
[16]	**ADR_MODE**	**Address mode control bit** 0 = 7-bit mode 1 = 10-bit mode **Note:** This bit is available in slaver mode only.	R/W	0x0
[15:8]	**ADDR2**	**Address 2nd Byte** 7-bit addressing mode : none 10-bit addressing mode : slave address[7:0]	R/W	0x0
[7:1]	**ADDR1[6:2]**	**Address 1st 7Bits** 7-bit addressing mode : ADDR1[6:0] = slave address[6:0] 10-bit addressing mode : ADDR1[6:2] = 0x1E ; ADDR1[1:0] = slave address[9:8]	R/W	0x61
[0]	**R_W**	I2C read / write control signal 0 = write 1 = read **Note:** This bit is shared between master and slaver mode.	R/W	0x0

I2C0 Data Register

I2C0_DATA **I2C 0 Data Register** **Address : 0x400B 000C**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I2C_DATA							

Bit	Name	Description	Access	Reset value
[31:8]	--	**Reserved**	R/W	0x00 0000
[7:0]	**I2C_DATA**	**I2C controller read / write data register** **Note:** The bit is shared by the master and slave modes.	R/W	0x00

8.11. Watchdog Timer

8.11.1. Overview

The independent watchdog (WDG) is clocked by its own dedicated low-speed clock (LSIRC32K) and thus stays active even if the main clock fails.

The WDG is best suited to applications which require the watchdog to run as a totally independent process outside the main application but has lower timing accuracy.

8.11.2. Features

●Free-running down-counter

●Clocked from an independent low speed RC oscillator

●Reset (if watchdog activated) when the down-counter value of 0x000 is reached

8.11.3. Function description

When the independent watchdog is started by writing the value 0x99 or 0x66 in the Key register (**WDG_KEY.KEYCODE**), the counter starts counting down from the reset value of 0xFFF. When it reaches the end of count value (0x000) a reset signal is generated (WDG reset).

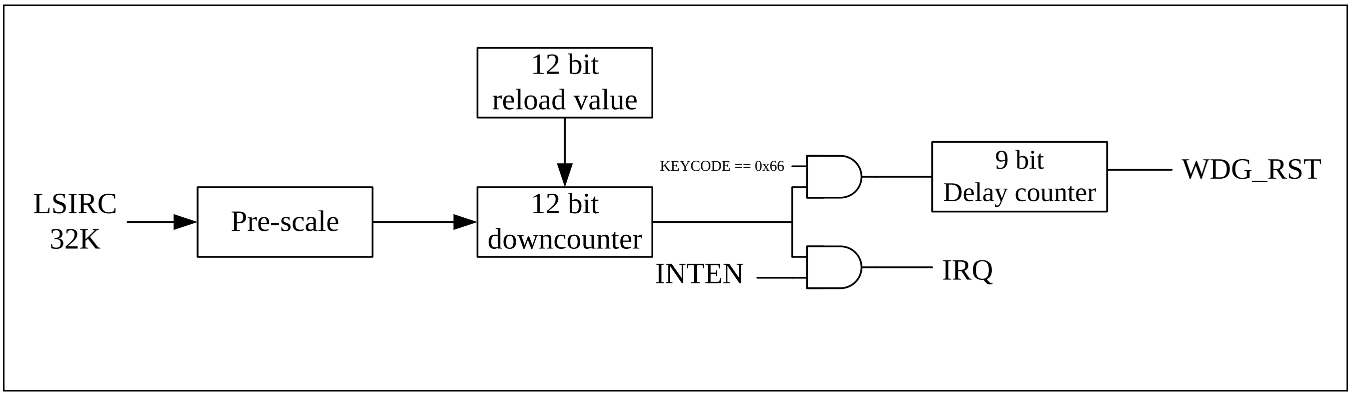
Whenever the key value 0xAA is written in the **KEYCODE** register, the **WDG_CTRL.RCNT** value is reloaded in the counter and the watchdog reset is prevented.

8.11.4. Register access protection

Write access to the **WDG_CTRL** register is protected. To modify it, user must first write the code 0x55 in the **WDG_CTRL** register. A write access to this register with a different value will break the sequence and register access will be protected again.

This implies that it is the case of the reload operation (writing 0xAA).

8.11.5. Block diagram



Register Map

Base Address : 0x400D_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
WDG_KEY	Watchdog Key Register	0x000	R/W	0x0000 0000
WDG_CTRL	Watchdog Control Register	0x004	R/W	0x0FFF 0000

Registers

Watchdog Key Register

WDG_KEY **Watchdog Key Register** **Address : 0x400D 0000**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEYCODE							

Bit	Name	Description	Access	Reset value
[31:8]	--	**Reserved**	R	0
[7:0]	**KEYCODE**	**Watchdog Key Code** AA = Reload the watchdog down counter value from **WDG_CTRL.RCNT** 55 = Enable access to WDG_CTRL 99 = Start the Watchdog counting without reset function 66 = Start the Watchdog counting with reset function 00 = Disable Watchdog function	R/W	0

Watchdog Control Register

WDG_CTRL **Watchdog Control Register** **Address : 0x400D 0004**

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**
--				**RCNT**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RCNT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--					**PRESCALE**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
--		**WDG_WORST**	**WDG_WIRST**	**--**	**WDG_EVEN**	**WDG_IFLG**	**WDG_INTEN**

Bit	Name	Description	Access	Reset value
[31:28]	**--**	**Reserved**	R	0
[27:16]	**RCNT**	**Watchdog counter reload value**	R/W	0xFFF
[15:11]	**--**	**Reserved**	R	0
[10:8]	**PRESCALE**	**Watchdog clock pre-scaler** 000 = LSIRC/1 001 = LSIRC/4 010 = LSIRC/8 011 = LSIRC/16 100 = LSIRC/32 101 = LSIRC/64 110 = LSIRC/128 111 = LSIRC/256	R/W	111
[7:6]	**--**	**Reserved**	R	0
[5]	**WDG_WORST**	**Watchdog running without reset function** 0 = Disabled 1 = Enabled Note: This bit effect based on WDG_KEY	R	0
[4]	**WDG_WIRST**	**Watchdog running with reset function** 0 = Disabled 1 = Enabled Note: This bit effect based on WDG_KEY	R	0
[3]	**--**	**Reserved**	R	0
[2]	**WDG_EVEN**	**Watchdog event Enable** 0 = Disabled 1 = Enabled Note: When RCNT down-count to zero, watchdog sends event to CPU. Note: It can wake up CPU from sleep mode via WFE.		
[1]	**WDG_IFLG**	**Watchdog timer timeout Flag** 0 = RCNT not down-count to zero. 1 = RCNT down-count to zero. Note: It's cleared by writing 1.	R/W	0
[0]	**WDG_INTEN**	**Watchdog Interrupt Enable** 0 = Disable. 1 = Enable. Note : When RCNT down-count to zero, watchdog sends IRQ to CPU.	R/W	0

8.12. WPC (Wireless Power Controller) RX

8.12.1. Overview

A wireless power system can be controlled easily with GPMQ9103A. Wireless Power Controller RX built-in several control register to make communication signals from RX to TX and modulated by ASK. And receive TX communication by FSK modulation. The communication method, data rates and data packet formats are defined by the Qi standard and be implement in these control registers.

Register Map

Base Address : 0x4016_0000				
Name	**Description**	**Offset Address**	**Access**	**Reset value**
WPRX_KEY	Wireless Power RX Key Register	0x000	R/W	0x0000 0000
WPRX_CTRL	Wireless Power RX Control Register	0x004	R/W	0x0000 0000
WPRX_INT	Wireless Power RX Interrupt Register	0x008	R/W	0x0000 0000
WPRX_STS	Wireless Power RX Status Register	0x00C	R	0x0000 0000

Registers

Wireless Power RX Key Register

WPRX_KEY Wireless Power RX Key Register Address : 0x4016 0000

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
--							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
KEYCODE							

Bit	Name	Description	Access	Reset value
[31:8]	--	**Reserved**	R	0
[7:0]	**KEYCODE**	**Wireless Power RX Control Register Key Code** Write 0xA5 to WPRX_KEY, the WPRX_CTRL registers will be unlock, all bit fields can be accessed.	R/W	0

Wireless Power RX Control Register

WPRX_CTRL **Wireless Power RX Control Register** **Address : 0x4016 0004**

31	30	29	28	27	26	25	24
--							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
--							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FAULT_INT	--		CLAMP_ON_SEL	COMM_ON_SEL	COMM	CLAMP	SINK

Bit	Name	Description	Access	Reset value
[31:8]	--	**Reserved**	R	0
[7]	**FAULT_INT**	**Fault Interrupt Pin Switch Enable (Write Protected)** 0 = Pin $\overline{\text{INT}}$ pulled up to VDD18. 1 = Pin $\overline{\text{INT}}$ pulled up to Low.	R/W	0
[6:5]	--	**Reserved**	R	0
[4]	**CLAMP_ON_SEL**	**CLAMP1/2 Turn On Mode Selection (Write Protected)** 0 = Turn on CLAMP1/2 MOS by WPRX_CTRL.CLAMP directly 1 = Turn on CLAMP1/2 MOS when AC1/AC2 in 0v.	R/W	0
[3]	**COMM_ON_SEL**	**COMM1/2 Turn On Mode Selection (Write Protected)** 0 = Turn on COMM1/2 MOS when AC1/AC2 in 0v. 1 = Turn on COMM1/2 MOS by WPRX_CTRL.COMM directly	R/W	0
[3]	**COMM**	**COMM1/2 Pin Switch Enable (Write Protected)** 0 = Turn Off 1 = Turn On	R/W	0
[1]	**CLAMP**	**CLAMP1/2 Pin Switch Enable (Write Protected)** 0 = Turn Off 1 = Turn On	R/W	0
[0]	**SINK**	**SINK Pin Switch Enable (Write Protected)** 0 = Turn Off 1 = Turn On	R/W	0

Wireless Power RX Interrupt Register

WPRX_INT Wireless Power RX Interrupt Register Address : 0x4016 0008

31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
FSK_TMO_FLG	CHG_FLG_OVR	RECT_OVP_FLG	VOUT_CHG_FLG	VOUT_OCL_FLG	VOUT_OV_FLG	OTP140_FLG	OTP130_FLG
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
FSK_TMO_INTE	--	RECT_OVP_INTE	VOUT_CHG_INTE	VOUT_OCL_INTE	VOUT_OV_INTE	OTP140_INTE	OTP130_INTE

Bit	Name	Description	Access	Reset value
[31:23]	--	**Reserved**	R	0
[23]	**FSK_TMO_FLG**	**FSK Timeout Flag** 0 = None 1 = AC1/AC2 don't active over 8ms. **Note:** Write 1 to clear this bit.	R/W	0
[22]	**CHG_FLG_OVR**	**VOUT_CHG_FLG Flag Override** 0 = None. 1 = VOUT switch ON/OFF changed when VOUT_CHG_FLG is high. **Note:** Write 1 to clear this bit.	R/W	0
[21]	**RECT_OVP_FLG**	**Rectifier Over Voltage Flag** 0 = None 1 = Rectifier voltage > RECT_OVP occurs. **Note:** Write 1 to clear this bit.	R/W	0
[20]	**VOUT_CHG_FLG**	**VOUT Voltage Change Flag** 0 = None 1 = VOUT switch ON/OFF changed. **Note:** Write 1 to clear this bit.	R/W	0
[19]	**VOUT_OCL_FLG**	**VOUT Over Current Limit Flag** 0 = None 1 = VOUT Source Current > LDOVOUT_CL occurs. **Note :** Write 1 to clear this bit.	R/W	0
[18]	**VOUT_OV_FLG**	**VOUT Over Voltage Flag** 0 = None. 1 = VOUT Voltage > (LDOVOUT_SEL * 1.25) occurs **Note:** Write 1 to clear this bit.	R/W	0
[17]	**OTP140_FLG**	**Over Temperature 140℃ Flag** 0 = None 1 = Die Temperature T > 140℃ occurs **Note:** Write 1 to clear this bit.	R/W	0
[16]	**OTP130_FLG**	**Over Temperature 130℃ Flag** 0 = None. 1 = Die Temperature T > 130℃ **Note:** Write 1 to clear this bit.	R/W	0
[15:8]	--	**Reserved**	R	0
[7]	**FSK_TMO_INTE**	**FSK Timeout Interrupt Enable** 0 = Disabled		

Bit	Name	Description	Access	Reset value
		1 = Enabled		
[6]	--	**Reserved**	R	0
[5]	**RECT_OVP_INTE**	**Rectifier Over Voltage Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0
[4]	**VOUT_CHG_INTE**	**VOUT Voltage Change Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0
[3]	**VOUT_OCL_INTE**	**VOUT Over Current Limit Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0
[2]	**VOUT_OV_INTE**	**VOUT Over Voltage Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0
[1]	**OTP140_INTE**	**Over Temperature 140℃ Protect Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0
[0]	**OTP130_INTE**	**Over Temperature 130℃ Protect Interrupt Enable** 0 = Disabled 1 = Enabled	R/W	0

Wireless Power RX Status Register

WPRX_STS	Wireless Power RX Status Register					Address : 0x4016 000C	
31	30	29	28	27	26	25	24
--							
23	22	21	20	19	18	17	16
--							
15	14	13	12	11	10	9	8
--							
7	6	5	4	3	2	1	0
--		RECT_OVP	VOUT_STS	VOUT_OCL	VOUT_OV	OTP140	OTP130

Bit	Name	Description	Access	Reset value
[31:6]	--	**Reserved**	R	0
[5]	**RECT_OVP**	**Rectifier Over Voltage Status** 0 = Rectifier voltage < RECT_OVP 1 = Rectifier voltage > RECT_OVP	R	0
[4]	**VOUT_STS**	**VOUT Status** 0 = VOUT OFF 1 = VOUT ON	R	0
[3]	**VOUT_OCL**	**VOUT Over Current Limit** 0 = VOUT Source Current < LDOVOUT_CL 1 = VOUT Source Current > LDOVOUT_CL	R	0
[2]	**VOUT_OV**	**VOUT Over Voltage** 0 = VOUT Voltage < (LDOVOUT_SEL * 1.25) 1 = VOUT Voltage > (LDOVOUT_SEL * 1.25)	R	0
[1]	**OTP140**	**Over Temperature 140℃ Protect** 0 = Die Temperature T < 120℃ 1 = Die Temperature T > 140℃	R	0
[0]	**OTP130**	**Over Temperature 130℃ Protect** 0 = Die Temperature T < 114℃ 1 = Die Temperature T > 130℃	R	0

9. ELECTRICAL CHARACTERISTICS

9.1. Absolute maximum ratings

Stresses above the values listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

Symbol	Parameter	Conditions	Min.	Max.	Unit
AC1/2, COMM1/2, CLAMP1/2	Absolute Maximum Pin Voltage		-0.8	24	V
/EN	Absolute Maximum Pin Voltage		-0.3	28	V
SINK, VRECT	Absolute Maximum Pin Voltage		-0.3	24	V
VDD18	Absolute Maximum Pin Voltage		-0.3	2	V
/INT, SCK, SDA, VDD5V, NTC, GPIO[4:0]	Absolute Maximum Pin Voltage		-0.3	6	V
BST1/2	Absolute Maximum Pin Voltage		-0.3	26	V
VOUT	Absolute Maximum Pin Voltage		-0.3	20	V
IOUT	Maximum Current on Pin			2.5	A
I_{SINK}, $I_{CLAMP1,}$ I_{CLAMP2}	Maximum Current on Pin			1	A
I_{COMM1}, I_{COMM2}	Maximum RMS Current on Pin			0.5	A
I_{AC1},I_{AC2}	Maximum RMS Current from Pin			2	A

9.1.1. Electrostatic discharge ratings

Electrostatic discharges are applied to the pins of each sample according to each pin combination. This test conforms to the JESDA22-A114/C101 standard.

Symbol	Parameter	Conditions	Min.	Max.	Unit
V_{ESD}	Electrostatic Discharge Voltage	Human Body Model; All Pins		2000	V
		Machine Model		200	V
		Charged-Device Model		500	V
I_{LATCH}	I/O Latch-Up Current	(0.5VDD) < V_{IN} < (1.5VDD)		100	mA

9.2. Operation conditions

The following operating conditions must not be exceeded in order to ensure correct operation and reliability of the GPMQ9103A. All parameters specified in the following tables refer to these operating conditions, unless noted otherwise.

Symbol	Parameter	Conditions	Min.	Max.	Unit
VDD	Standard operating voltage		3	5.5	V
f_{HCLK}	Internal AHB clock frequency		0	48	MHz
f_{PCLK}	Internal APB clock frequency		0	48	MHz
T_A	Ambient Temperature		-40	85	˚c
T_J	Junction Temperature		-40	150	˚c
T_{STG}	Storage Temperature		-65	150	˚c

9.3. Power-Up and Reset Characteristics

T_A = -40˚c to 85˚c , unless otherwise specified.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
$VRECT_{UVLO}$(1)	Under-Voltage Lock-Out			2.8		V

(1) Guaranteed by characterization results, not tested in production.

9.4. Electrical characteristics

T_A = -40˚c to 105˚c; VRECT = 2.4v to 5.5v; COUT = 4.7µF, /EN = LOW; Typical values are at T_A = 25˚c unless otherwise specified.

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
Rectifier						
VRECT UVLO	**Under-Voltage Lock-Out**	VRECT ramps from 0 to 4V	2.5	2.8	3	V
VRECT HYS	UVLO Hysteresis			0.3		V
VRECT OVP	VRECT Over Voltage Protect	Selectable : 11/13/15/17	-5%		5%	V
VRECT HYS	VRECT OVP Hysteresis			1		V
Ron	Rectifier Switch Rds ON			35		mΩ
C of CLAMP1/2				0.47		uF
C of COMM1/2				0.022		uF
C of BST1/2			0.01	0.015		uF
C of VRECT			20	30		uF
Power Switch / LDO						
ILIM	Current limit level	Selectable : 1A, 1.5A, 2A	-10%	1 ~ 2	10%	A
IOUT	Maximum Output Current			1.67		A
VOUT	Output voltage	Selectable : 5/7/9/12v, -2%	-3%	5 ~ 12	+3%	V
VOUT OVP	VOUT Over Voltage Protect			VOUT+25%		V
V5P0 LDO						
V5P0 LDO		Selectable : 4.5/5.0/5.2/5.5	-5%	4.5 ~ 5.5	5%	V
C_{V5P0}			0.5	1	4.7	uF
V1P8 LDO						
V1P8 LDO			1.62	1.8	1.98	V
C_{V1P8}			0.5	1	4.7	uF
Quiescent						
ISHD	Shot down mode current	/EN = H, VRECT=12.3v		500		uA
IACT	Supply current	/EN = L, VRECT=12.3v, No load		3		mA

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
Temperature Sensor						
Avg_Slope	Average slope			1.625		mV/˚c
V25	Voltage at 25 °C			0.752		V
T_{W}	Thermal Warning			130		˚C
T_{SD}	Thermal Shut-Down			140		˚C
GPIO						
V_{IL}	Standard IO input low level voltage				0.7	V
V_{IH}	Standard IO input high level voltage		1.4			V
V_{OL}	Standard IO output low level voltage	I_{LOAD} = 8mA			0.36	V
R_{PU}	Weak pull-up equivalent resistor			50		KΩ
R_{PD}	Weak pull-down equivalent resistor			50		KΩ
I_{LKG}	Input leakage current		-1		1	μA
Reset Pin						
V_{IL}	Standard IO input low level voltage				0.7	V
V_{IH}	Standard IO input high level voltage		1.4			V
R_{PU}	Weak pull-up equivalent resistor	VDD = 3.3v, VIN = VSS		50		KΩ
I_{LKG}	Input leakage current	VSS ≤ VIN ≤ VDD	-1		1	μA
10Bit ADC						
VDD_{ADC}	ADC supply voltage		3		5.5	V
V_{ADCREF}	ADC reference voltage		-1%	2.1	+1%	V
V_{ADCIN}	Conversion voltage range		0		V_{ADCREF}	V
Channel	ADC measure channel			12		Ch
RES_{ADC}	Resolution			10		Bit
f_{ADC}	ADC clock frequency			1.5		MHz
f_{SAMP}	Sampling rate	ts = $1/f_{ADC}$		107		KSPS
Rint	Sampling switch resistance			0.5		KΩ
C_{S}	Internal sample and hold capacitor			0.1		pF
Ts	Sampling time		1		200	$1/f_{ADC}$
tconv	Total conversion time (including sampling time)	F_{ADC} = 1.5MHz	9.33		142	μS
			14 to 213 (ts for sampling +13 for successive approximation)			$1/f_{ADC}$
EO	Offset error		-3		+3	LSB
ED	Differential linearity error		-0.5		+0.5	LSB
EL	Integral linearity error		-1		+1	LSB
IOSC 48M						
f_{HSIRC}	Internal high speed RC frequency			48		MHz
Duty	Duty cycle		45	50	55	%
ACC_{HSIRC}	Accuracy of the HS oscillator		-1.5		1.5	%
IOSC 32K						
f_{LSIRC}	Internal low speed RC frequency			31.25		KHz

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
Duty	Duty cycle			50		%
ACC_{LSIRC}	Accuracy of the LS oscillator		-1.5		1.5	%

(1) Guaranteed by characterization results, not tested in production.

(2) The ADC is monotonic, there are no missing codes.

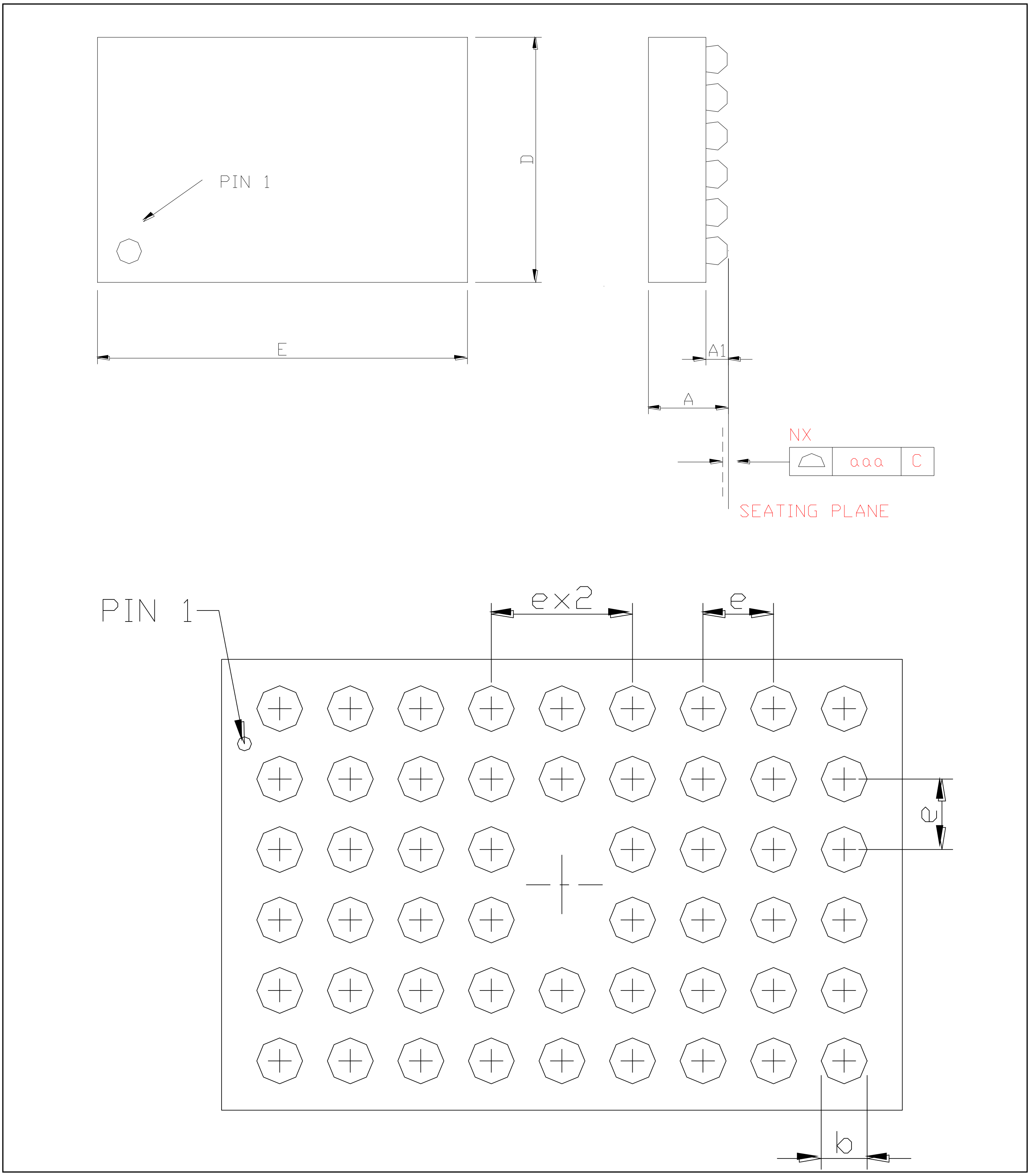
(3) EO = Offset Error: deviation between the first actual transition and the first ideal one.

(4) ED = Differential Linearity Error: maximum deviation between actual steps and the ideal one.

(5) EL = Integral Linearity Error: maximum deviation between any actual transition and the end point correlation line.

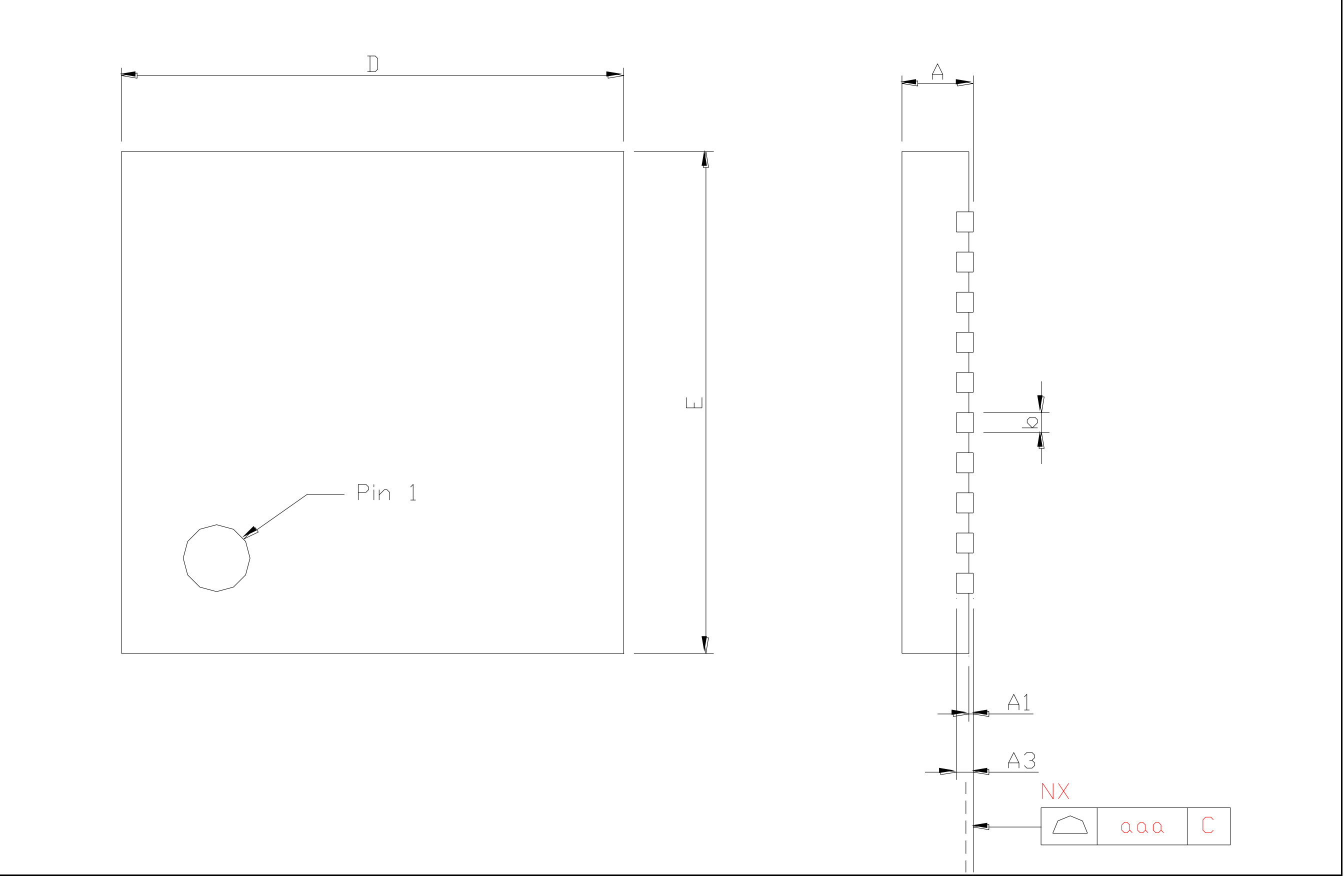
10. PACKAGE DIMENSIONS

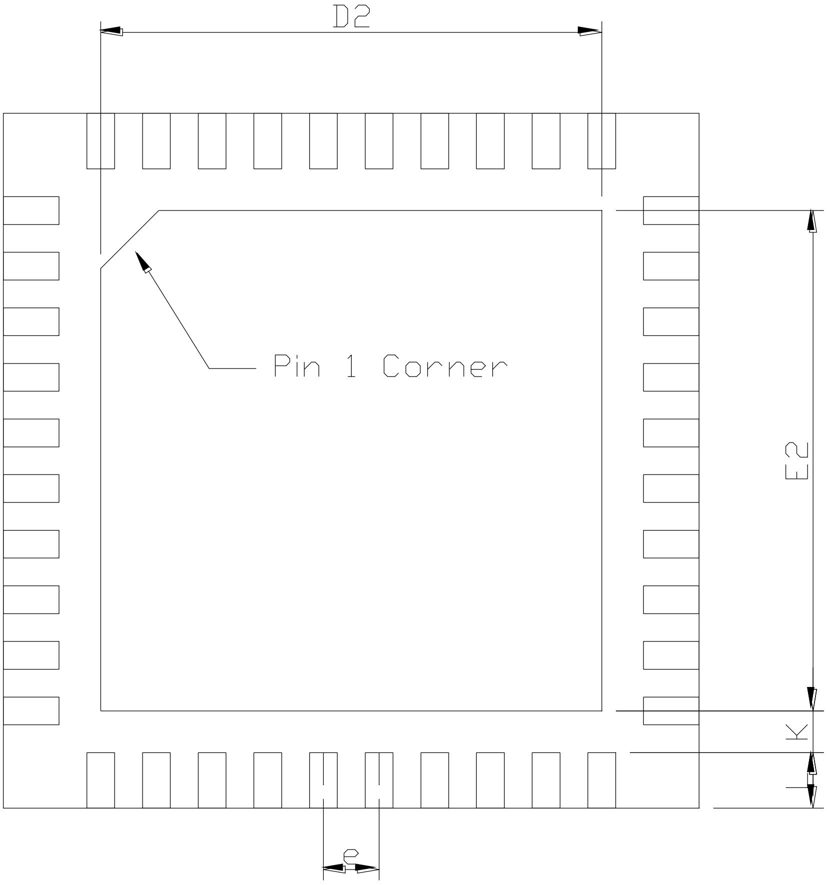
WLCSP (footprint)



SYMBOL	WLCSP2.56*3.86-52			
	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	0.55	0.65	0.022	0.026
A1	0.17	0.22	0.007	0.009
b	0.24	0.30	0.009	0.012
D	2.56	2.64	0.101	0.104
E	3.86	3.94	0.152	0.155
e	0.40 BSC		0.016 BSC	
aaa	0.08		0.003	

TQFN40 (5x5x mm footprint)





S Y M B O L	TQFN5*5-40B			
	MILLIMETERS		INCHES	
	MIN.	MAX.	MIN.	MAX.
A	0.70	0.80	0.028	0.031
A1	0.00	0.05	0.000	0.002
A3	0.20 REF		0.008 REF	
b	0.15	0.25	0.006	0.010
D	4.90	5.10	0.193	0.201
D2	3.50	3.70	0.138	0.146
E	4.90	5.10	0.193	0.201
E2	3.50	3.70	0.138	0.146
e	0.40 BSC		0.016 BSC	
L	0.35	0.45	0.014	0.018
K	0.20		0.008	
aaa	0.08		0.003	

11. DISCLAIMER

The information appearing in this publication is believed to be accurate.

Integrated circuits sold by Generalplus Technology are covered by the warranty and patent indemnification provisions stipulated in the terms of sale only. GENERALPLUS makes no warranty, express, statutory implied or by description regarding the information in this publication or regarding the freedom of the described chip(s) from patent infringement. FURTHERMORE, GENERALPLUS MAKES NO WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE. GENERALPLUS reserves the right to halt production or alter the specifications and prices at any time without notice. Accordingly, the reader is cautioned to verify that the data sheets and other information in this publication are current before placing orders. Products described herein are intended for use in normal commercial applications. Applications involving unusual environmental or reliability requirements, e.g. military equipment or medical life support equipment, are specifically not recommended without additional processing by GENERALPLUS for such applications. Please note that application circuits illustrated in this document are for reference purposes only.

12. REVISION HISTORY

Date	Revision #	Description	Page
Nov. 17, 2020	0.3	Modify some description in article. 1) WPC Qi compliant 2) The ADC_RDY status after ADCx_CTRL0.ADC_E set to 1	 5 111
Jan. 31, 2020	0.2	Modify some description in article. 1) Change VREC OVP Level 2) Remove I2C/ADC DMA function description 3) Remove function of FSK_MOD_HYS and change to CLAMP_ON_SEL & COMM_ON_SEL.	 59 142 161
Jun. 17, 2019	0.1	Preliminary version	