SWADESHI MICROPROCESSOR CHALLENGE-2020

SHAKTI SoC DEVICE REGISTER MANUAL

DEVELOPED BY :
SHAKTI DEVELOPMENT TEAM @ IITM
SHAKTI.ORG.IN

0.1 Proprietary Notice

The project was funded by Ministry of Electronics and Information Technology (MeitY), Government of India

0.2 Release Information

Version	Date	Updates
1.0	02/10/2020	Initial version
1.1	12/11/2020	corrections
1.2	27/01/2021	Addition of CLINT

Table of Contents

PULSE WIDTH MODULATORS

Pulse Width Modulators (PWM) are used to generate pulses with variable duty cycle. The duty cycle and the period of the pulse can be varied through DUTY and PERIOD registers respectively.

PWM Registers

Register name	Offset address	Accessible Size	Description
PERIOD_REGISTER	‘h00	16 bits	PWM Period Register (Read and Write)
DUTY_REGISTER	‘h04	16 bits	PWM Duty Register (Read and Write)
CONTROL_REGISTER	‘h08	8 bits	PWM Control Register (Read and Write)
CLOCK_REGISTER	‘h0C	16 bits	PWM Clock Register (Read and Write)

PWM CONTROL Register

PWM can be configured and controlled with this register.

7	6	5	4	3	2	1	0
RESET_COUNTER	Reserved	INTERRUPT	PWM_OUTPUT ENABLE	CONTINUOUS_ONCE	PWM_ START	PWM_ENABLE	CLOCK_SELECT

Signal name	**Bit position**	**Comments**
RESET_COUNTER	7	PWM Reset
Reserved	6	Reserved
INTERRUPT	5	0: Interrupt not Occured. 1: Interrupt occurred.
PWM_OUTPUT_ENABLE	4	0: Disable PWM output 1: Enable PWM output
CONTINUOUS_ONCE	3	0: In timer mode, continuous mode is Off. 1: In timer mode, continuous mode is ON.
PWMSTART	2	1:Start PWM Operation
PWM ENABLE	1	0: Timer mode enable 1: PWM enable
CLOCK_SELECT	0	0: Internal clock source selected 1:External clock source selected

PWM CLOCK_REGISTER

PWM CLOCK_REGISTER divides the system clock by the (Prescaler value + 1).

15	1	0
PWM PRESCALER		Reserved

Bits 15:1 - PWM PRESCALER : Prescale value to get required clock speed.
PWM clock frequency = Internal clock frequency / (Prescale + 1).

PWM PERIOD_REGISTER

PWM PERIOD_REGISTER further divides the system clock by the (Prescaler value + 1).

15	0
PWM Period	

Bits 15:1 - PWM PERIOD : Period value to get required PWM period..

Required PWM Period = PWM clock frequency / (Period + 1).

PWM DUTY_REGISTER

PWM DUTY_REGISTER decides the ON time of PWM clock based on the duty value..

15	0
PWM Duty	

Bits 15:1 - PWM DUTY: ON Time of the PWM Period..

ON (HIGH) of PWM DUTY = PWM Period / (Duty + 1) .

SPI

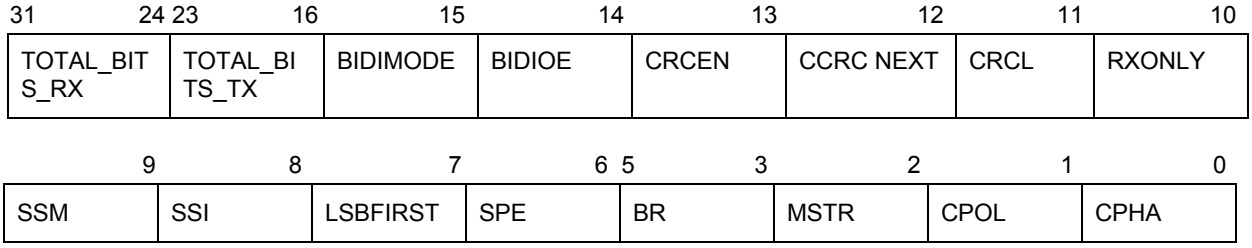
SPI is a synchronous serial I/O port that allows a serial bit stream of programmed length to be shifted into and out of the device at programmable bit transfer rate.

SPI Registers

Register name	Offset address	Accessible Size	Description
SPI_CR1	‘h00	32 bits	SPI Control Register 1 (Read and Write)
SPI_CR2	‘h04	32 bits	SPI Control Register 2 (Read and Write)
SPI_SR	‘h08	32 bits	SPI Status Register (Read Only)
SPI_DR1	‘h0C	32 bits	SPI Data Register 1 (Read and Write)
SPI_DR2	‘h10	32 bits	SPI Data Register 2 (Read and Write)
SPI_DR3	‘h14	32 bits	SPI Data Register 3 (Read and Write)
SPI_DR4	‘h18	32 bits	SPI Data Register 4 (Read and Write)
SPI_DR5	‘h1C	32 bits	SPI Data Register 5 (Read and Write)
SPI_CRCPR	‘h20	32 bits	Reserved
SPI_RXCRCR	‘h24	32 bits	Reserved
SPI_TXCRCR	‘h28	32 bits	Reserved

SPI_CR1 Register

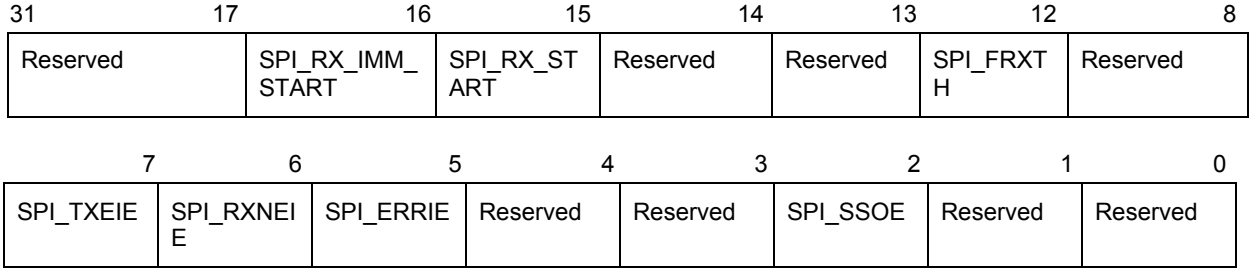
SPI_CR1 and SPI_CR2 configures and controls the SPI for operation.



Signal Name	Bit Position	Comments
SPI_TOTAL_BITS_RX	31:24	Expected Total Number of bits to be received.
SPI_TOTAL_BITS_TX	23:16	Total Number of bits to be transmitted.
SPI_BIDIMODE	15	Bidirectional data mode enable. This bit enables half-duplex communication using a common single bidirectional data line. 0: 2-line unidirectional data mode selected 1: 1-line unidirectional data mode selected
SPI_BIDIOE	14	Output enable in bidirectional mode This bit combined with the BIDI-MODE bit selects the direction of transfer in bi-direction mode. 0: receive-only mode (Output Disabled) 1: transmit-only mode (Output Enabled)
SPI_CRCEN	13	:Hardware CRC calculation Enable. 0: CRC calculation disable 1: CRC calculation enable
SPI_CCRCNEXT	12	Transmit CRC Next. 0: Next Transmit value is from Tx buffer 1: Next Transmit value is from Rx buffer
SPI_CRCL	11	CRC Length bit is set and cleared by software to select CRC Length
SPI_RXONLY	10	Receive only mode enabled. This bit enables simplex communication using a single unidirectional line to receive data exclusively. Keep BIDIMODE bit clear when receiving the only mode is active.

SPI_SSM	9	Software Slave Management. When the SSM bit is set, the NSS pin input is replaced with the value from the SSI bit. 0: Software slave management disabled 1: Software slave management enabled
SPI_SSI	8	Internal Slave Select.This bit has an effect only when the SSM bit is set. The value of this bit is forced onto the NSS pin and the I/O value of the NSS pin is ignored
SPI_LSBFIRST	7	Frame Format 0: data is transmitted/received with the MSB first 1: data is transmitted/received with the LSB first Note:This bit should not be changed when communication is ongoing
SPI_SPE	6	SPI Enable 0: SPI is disabled 1: SPI is enabled
SPI_BR[2:0]	5:3	Baud Rate Control 000: fCLK/2 001: fCLK/4 010: fCLK/8 011: fCLK/16 100: fCLK/32 101: fCLK/64 110: fCLK/128 111: fCLK/256 Note:This bit should not be changed when communication is ongoing
SPI_MSTR	2	Master Selection 0: Slave Configuration 1: Master Configuration Note This bit should not be changed when communication is ongoing
SPI_CPOL	1	Clock Polarity 0: CLK is 0 when idle 1:CLK is 1 when idle
SPI_CPHA	0	Clock Phase 0: The first clock transition is the first data capture edge 1: The second clock transition is the first data capture edge

SPI_CR2 Register



Signal Name	Bit Position	Comments
Reserved	31:17	Reserved
SPI_RX_IMM_START	16	Set this to receive data immediately after transmission finished. This should be set before configuring the control register 1. 0: Immediate Receive data disabled 1: Immediate Receive data enabled
SPI_RX_START	15	Set this to start receive data only. This is similar to RX_ONLY in control register 1. 0: Receive only disabled 1: Receive only enabled
Reserved	14	Reserved
Reserved	13	Reserved
SPI_FRXTH	12	FIFO reception threshold is used to set the threshold of the RXFIFO that triggers an RXNE event. 0: RXNE event is generated if the FIFO level is greater than or equal to 1/2 (16-bit) 1: RXNE event is generated if the FIFO level is greater than or equal to 1/4 (8-bit)
Reserved	11:8	Reserved
SPI_TXEIE	7	Interrupt enable for TXE event. 0: TXE interrupt masked 1: TXE interrupt is not interrupt masked
SPI_RXNEIE	6	Interrupt enable for RXNE event 0: RXNE interrupt masked 1: RXNE interrupt is not interrupt masked

SPI_ERRIE	5	Error interrupt enable bit controls the generation of interrupt when an error condition occurs. 0: Error interrupt masked 1: Error interrupt not masked.
Reserved	4	Reserved
Reserved	3	Reserved
SPI_SSOE	2	SS output enable 0: SS output is disabled in master mode and the SPI interface can work in a multi-master configuration 1: SS output is enabled in master mode and when the SPI interface is enabled. The SPI interface cannot work in a multi-master environment.
Reserved	1	Reserved
Reserved	0	Reserved

SPI_SR Register

SPI_SR register provides the various status of the SPI transmit and receive operations.
Address offset: 0x08

31 13	12 11	10 9	8	7	6	5	4	3 2	1	0
Reserved	SPI_FTLVL	SPI_FRLVL	SPI_FRE	SPI_BSY	SPI_OVR	SPI_MODF	SPI_CRCERR	Reserved	SPI_TXE	SPI_RXNE

Signal Name	Bit Position	Comments
Reserved	31:13	Reserved
SPI_FTLVL	12:11	FIFO Transmission Level. These bits are set and cleared by hardware. 00: FIFO empty 01: 1/4 FIFO 10: 1/2 FIFO 11: FIFO full

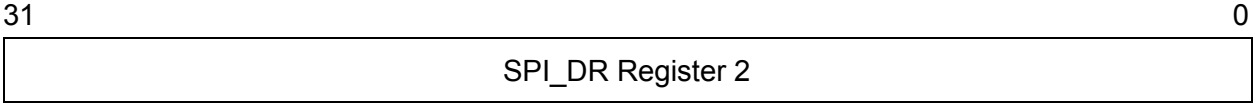
SPI_FRLVL	10:9	FIFO reception level.These bits are set and cleared by hardware. 00: FIFO empty 01: 1/4 FIFO 10: 1/2 FIFO 11: FIFO full
SPI_FRE	8	Frame format error. This flag is used for SPI in the TI slave mode. This flag is set by hardware and reset when SPIx SR is read by software. 0: No frame format error 1: A frame format error occurred
SPI_BSY	7	The BSY flag is set and cleared by hardware (writing to this flag has no effect). When BSY is set, it indicates that a data transfer is in progress on the SPI (the SPI bus is busy). 0: SPI not busy 1: SPI is busy in communication or Tx buffer is not empty
SPI_OVR	6	Overrun condition
SPI_MODF	5	Mode fault occurs when the master device has its internal NSS signal (NSS pin in NSS hardware mode, or SSI in NSS software mode) pulled low. 0: No mode fault occurred 1: Mode fault occurred
SPI_CRCERR	4	CRC error flag 0: CRC value received matches the SPI_RXCRCR value 1: CRC value received does not match the SPI_RXCRCR value
Reserved	3	Reserved
Reserved	2	Reserved
SPI_TXE	1	This flag is set when transmission TXFIFO has sent all the data
SPI_RXNE	0	The RXNE flag is set when RX buffer finishes transferring data to register

SPI_DR1 Register

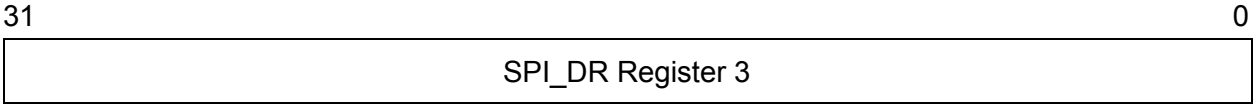
SPI_DR Register (DR1 - DR5) holds the data value that can be transmitted and received SPI_DR. When transmitting data, the data needs to be filled from SPI_DR1 to SPI_DR5. The 31

- 24 should have first byte, Bits 23 - 16 should have second byte and so on. Once the required data bytes are filled. Finally SPI_DR5 should be written with some (dummy) value.

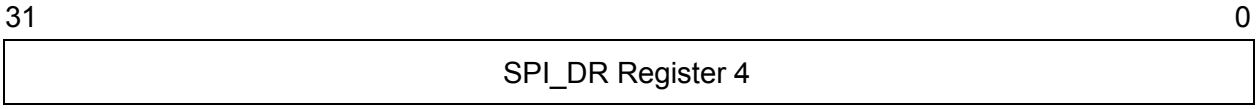
SPI_DR2 Register



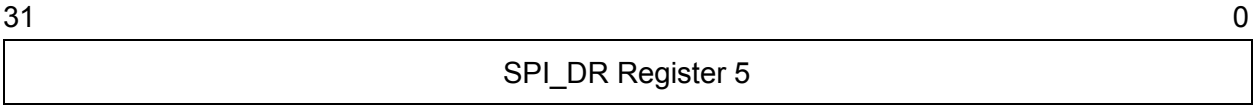
SPI_DR3 Register



SPI_DR Register 4



SPI_DR Register 5



I2C

I2C is a serial protocol for a two-wire interface to connect low-speed devices like microcontrollers, EEPROMs, A/D and D/A converters, I/O interfaces and other similar peripherals in embedded systems. Multiple slave devices can be connected to a single master with I2C. I2C only uses two wires to transmit data between devices

I2C Registers

Register name	Offset address	Accessible Size	Description
Prescale	'h00	8 bits	I2C Prescaler Register (Read and Write)
Control	'h08	8 bits	I2C Control Register (Read and Write)
Data	'h10	8 bits	I2C Data Shift Register (Read and Write)
Status	'h18	8 bits	I2C Status Register (Read)
SCL	'h38	8 bits	I2C Clock Register (Read and Write)

I2C PRESCALE Register

I2C Prescale Register divides the System clock by (Prescale value + 1). This clock is used as clock input for I2C Serial Clock.
I2C Prescaler clock = System Clock / (Prescaler Value + 1)

Address offset: 0x0

7	6	5	4	3	2	1	0
PRESCALE VALUE							

I2C SCL Register

I2C SCL Register divides the I2C Prescaler clock by (SCL value + 1). This clock is used as
I2C SCL clock = I2C Prescaler Clock / (SCL COUNT + 1).

Address offset: 0x38

7	6	5	4	3	2	1	0
SCL COUNT							

Status Register

I2C Status Register has status of I2C data transfer. The bit level details are given below.

Address offset: 0x18

7	6	5	4	3	2	1	0
I2C_INI	Reserved	I2C_STS	I2C_BER	I2C_AD0/i I2C_LRB	I2C_AAS	I2C_LAB	I2C_BB

Signal Name	Bit Position	Comments
I2C_INI	7	High when I2C communication in progress. Becomes low once I2C communication is complete.
Reserved	6	Set to 0
I2C_STS	5	When in slave receiver mode, this flag is asserted when an externally generated STOP condition is detected (used only in slave receiver mode).
I2C_BER	4	Bus error; a misplaced START or STOP condition has been detected
I2C_AD0/I2C_ LRB	3	AD0(Address 0) - General Call bit used for Broadcast LRB - Last Received Bit through I2C Bus This status bit serves a dual function, and is valid only while PIN = 0: 1. LRB holds the value of the last received bit over the I2C-bus while AAS = 0 (not addressed as slave). Normally this will be the value of the slave acknowledgement; thus checking for slave acknowledgement is done via testing of the LRB. 2. AD0; when AAS = 1 (‘Addressed As Slave’ condition), the I2C-bus controller has been addressed as a slave. Under this condition, this bit becomes the ‘AD0’ bit and will be set to logic 1 if the slave address received was the ‘general call’ (00H) address, or logic 0 if it was the I2C-bus controller’s own slave address.

I2C_AAS	2	Addressed As Slave' bit. Valid only when PIN = 0. When acting as slave receiver, this flag is set when an incoming address over the I2C-bus matches the value in own address register
I2C_LAB	1	Lost Arbitration Bit. This bit is set when, in multi-master operation, arbitration is lost to another master on the I2C-bus.
I2C_BB	0	Bus Busy bit. This is a read-only flag indicating when the I2C-bus is in use. A zero indicates that the bus is busy, and access is not possible

Control Register

I2C control register configures and controls the I2C data transfer

Address offset: 0x08

7	6	5	4	3	2	1	0
I2C_PIN	I2C_ES0	Reserved	Reserved	I2C_ENI	I2C_STA	I2C_STO	I2C_ACK

Signal Name	**Bit Position**	**Comments**
SPI_PIN	7	Pending Interrupt Not, Used as a software reset. If set to 1, all
SPI_ESO	6	Enable Serial Output. 0 - Registers can be initialized. 1 - I2C Serial Transmission
Reserved	5	Reserved.
Reserved	4	Reserved.
SPI_ENI	3	Enables the external interrupt output, which is generated when the PIN is active (Low)
SPI_STA	2	Transmits Start condition + Slave address..
SPI_STO	1	Transmits the stop condition.
SPI_ACK	0	Acknowledgement bit. 1: I2C automatically sends an acknowledgement after a read/write transaction. 0: I2C Master sends Negative Acknowledge to stop the I2C transfer.

UART

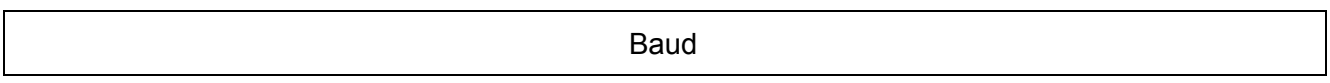
UART module provides a two-wire asynchronous serial non-return-to-zero (NRZ) communication with RS-232 (RS-422/485) interface. Each UART module has transmit and receive buffers that can hold upto 16 entries. Data transfer rate can be modified by providing appropriate value to UARTBAUD register.

UART Registers

Register name	Offset address	Accessible Size	Description
BAUD Register	‘h00	16 bits	UART Baud Rate Register (Read and write)
TX_REG Register	‘h04	32 bits	UART Transmitter Data Register (Write only)
RCV_REG Register	‘h08	32 bits	UART Receiver Data Register (Read Only)
Status Register	‘h0C	8 bits	UART Status Register (Read only)
Delay Register	‘h10	16 bits	UART Transmitter Delay Register(Read and write)
Control Register	‘h14	16 bits	UART Control Register (Read and write)
IEN Register	‘h18	8 bits	UART Interrupt Enable Register (Read and write)
IQCYCLES Register	‘h1C	8 bits	UART Input Qualification Control Register (Read and write)
RX_THRESHOLD Register	‘h20	8 bits	UART Receiver Threshold Register (Read and write)

UART Baud Register

The UART BAUD Register configures the Baud rate based on the Baud value set.
Address offset: 0x00



Signal Name	Bit Position	Read/Write	Comments
Baud	15:0	Read/Write	Baud Rate is calculated by Baud_value=(Clock Frequency)/(16*Baudrate)

UART Status Register

The status register holds the status of various Transmit and receive status and errors.

7	6	5	4	3	2	1	0
BREAK_ERROR	FRAME_ERROR	OVERRUN	PARITY_ERROR	STS_RX_FULL	STS_RX_NOT_EMPTY	STS_TX_FULL	STS_TX_EMPTY

Signal Name	Bit Position	Comments
BREAK_ERROR	7	Break Error (Sets when the data and stop are both zero)
FRAME_ERROR	6	Frame Error (Sets when the stopis zero)
OVERRUN	5	Overrun Error (A data overrun error occurred in the receive shift register. This happens when additional data arrives while the FIFO is full.)
PARITY-ERROR	4	Parity Error (Sets when The receive character does not have correct parity information and is suspect.)
STS_RX_FULL	3	Receiver Full (Sets when the Receive Buffer is Full)
STS_RX_NOT_FULL	2	Receiver Not Empty (Sets when there is some data in the Receive Buffer).
STS_TX_FULL	1	Transmitter Full (Sets when the transmit Buffer is full)
STS_TX_EMPTY	0	Transmitter Empty(Sets when the Transmit Buffer is empty).

UART Control Register

The UART and receive operation and protocol are configured using the UART control register.

16 11	10 5	4 3	2 1	0
Reserved	UART_TX_RX_LEN	PARITY	STOP_BITS	Reserved

Signal Name	Position	Comments
Reserved	16:11	Reserved
UART_TX_RX_LEN	10:5	Character size of data. Maximum length is 32 bits.

PARITY	4:3	Insert Parity bits 00 - None 01 - Odd 10- Even 11 - Unused or Undefined
STOP_BITS	2:1	Stop bits 00 - 1 Stop bits 01 - 1.5 Stop bits 10 - 2 Stop bits
Reserved	0	Reserved

UART TX_REG Register

The value to be transmitted is written into the TX_REG register.
Address offset: 0x04

32	0
Data to be transmitted	

UART RCV_REG Register

The received value can be read from the UART Receive DATA register.
Address offset: 0x08

32	0
UART Receive Data	

UART IEN Register

Enables interrupts based on bit value set in the Interrupt Enable Register.
Address offset:0x18

8	7	6	5	4	3	2	1	0
ENABLE_RX_THRESHOLD	ENABLE_BREAK_ERROR	ENABLE_FRAME_ERROR	ENABLE_OVERRUN	ENABLE_PARITY_ERROR	ENABLE_RX_FULL	ENABLE_RX_NOT_EMPTY	ENABLE_TX_FULL	ENABLE_TX_EMPTY

Signal Name	Bit Position	Comments
Reserved	15:9	Reserved
ENABLE_RX_THRESHOLD	8	RX Threshold Interrupt Enable
ENABLE_BREAK_ERROR_EN	7	Break Error Interrupt Enable

ENABLE_FRAME_ERROR_EN	6	Frame Error Interrupt Enable
ENABLE_OVERRUN_ERROR_EN	5	Overrun Interrupt Enable
ENABLE_PARITY_ERROR_EN	4	Parity Error Interrupt Enable
ENABLE_RX_FULL	3	Receiver Full Interrupt Enable
ENABLE_RX_NOT_EMPTY	2	Receiver Not Empty Interrupt Enable
ENABLE_TX_FULL	1	Transmitter Full Interrupt Enable
ENABLE_TX_EMPTY	0	Transmission Empty Interrupt Enable

UART DELAY Register

Delayed Transmit control is done by providing the required delay in UART DELAY register.
Address offset: 0x10

16	0
Transmit Delay Count	

UART IQCYCLES Register

UART IQCYCLES Register holds the number of input qualification cycles for the receiver pin. .
Address offset: 0x1C

8	0
Input Qualification Cycles Count	

UART RX_THRESHOLD Register

UART RX_THRESHOLD register holds the receiver FIFO threshold value, when the RX FIFO level increases beyond the threshold, corresponding status bit will be set and when interrupt is enabled, interrupt will be raised
Address offset: 0x20

8	0
FIFO Receive Threshold Value	

GENERAL PURPOSE INPUT / OUTPUT (GPIO)

The General Purpose Input/output operation can be used to generate custom waveforms, enable signals, generate interrupts, etc. The GPIO has a GPIO DIRECTION register which configures the GPIO pin as an input or output and the GPIO DATA register which holds the input data to GPIO or output data from GPIO. The GPIO pins 0 - 7 can accept External events as interrupts. To use a GPIO pin (0 - 7) as interrupt, that particular GPIO pin(s) should be configured as input. The GPIO data register is 1 byte, 2 byte and 4 byte accessible.

GPIO Registers

Register name	Offset address	Accessible size	Description
GPIO_DIRECTION_ CNTRL_REG	‘h00	32 bits	GPIO Direction Control Register (Read and Write)
GPIO_DATA_REG	`h08	32, 16, 8 bits	GPIO Data Register (Read and Write)

Sequence of execution:

1. Write into the GPIO Direction register to configure GPIO pin as an input or output.
2. Write appropriate values to the GPIO DATA register.

Interrupt ID mapping to Device number

The following interrupts can be mapped with the respective interrupt IDs and the IDs are used when configuring PLIC registers to enable and service a particular interrupt.

ID	Source	ID	Source
0	Unused	15	GPIO8
1	PWM5	16	GPIO9
2	PWM4	17	GPIO10
3	PWM3	18	GPIO11
4	PWM2	19	GPIO12
5	PWM1	20	GPIO13
6	PWM0	21	GPIO14
7	GPIO0	22	GPIO15
8	GPIO1	23	I2C0
9	GPIO2	24	I2C1
10	GPIO3	25	UART0
11	GPIO4	26	UART1
12	GPIO5	27	UART2
13	GPIO6		
14	GPIO7		

Core Local Interrupter (CLINT)

The purpose of CLINT is to configure timer interrupts.

CLINT Registers

Register name	Offset address	Accessible Size	Description
mtime	‘hBFF8	64 bits	mtime register (Read and write)
mtimecmp	‘h4000	64 bits	mtime compare register (Read and write)

mtime Register

A memory mapped register of a real time counter.
Address offset: 0xBFF8

63	0
mtime	

mtimecmp Register

Machine mode timer compare register which causes a timer interrupt to be posted when the mtime register contains a value greater than or equal to the value in the mtimecmp register
Address offset: 0x4000

63	0
mtimecmp	

mtimecmp = mtime + N * (Clock frequency / mtime divisor)

N - Number of seconds after which the interrupt is generated.
Clock Frequency - Frequency at which device clock in running
mtime divisor - Value of this divisor varies as per board
Pinaka - 16
Parashu - 256
Vajra - 256