3 BSC

3.1 Introduction

The Broadcom Serial Controller (BSC) controller is a master, fast-mode (400Kb/s) BSC controller. The Broadcom Serial Control bus is a proprietary bus compliant with the Philips® I2C bus/interface version 2.1 January 2000.

- I^2C single master only operation (supports clock stretching wait states)
- Both 7-bit and 10-bit addressing is supported.
  - Timing completely software controllable via registers

3.2 Register View

The BSC controller has eight memory-mapped registers. All accesses are assumed to be 32-bit. Note that the BSC2 master is used dedicated with the HDMI interface and should not be accessed by user programs.

There are three BSC masters inside BCM. The register addresses starts from

- BSC0: 0x7E20_5000
- BSC1: 0x7E80_4000
- BSC2 : 0x7E80_5000

The table below shows the address of I^2C interface where the address is an offset from one of the three base addreses listed above.

I2C Address Map			
Address Offset	**Register Name**	**Description**	**Size**
0x0	C	Control	32
0x4	S	Status	32
0x8	DLEN	Data Length	32
0xc	A	Slave Address	32
0x10	FIFO	Data FIFO	32
0x14	DIV	Clock Divider	32
0x18	DEL	Data Delay	32

0x1c	CLKT	Clock Stretch Timeout	32

C Register

Synopsis The control register is used to enable interrupts, clear the FIFO, define a read or write operation and start a transfer.
The READ field specifies the type of transfer.
The CLEAR field is used to clear the FIFO. Writing to this field is a one-shot operation which will always read back as zero. The CLEAR bit can set at the same time as the start transfer bit, and will result in the FIFO being cleared just prior to the start of transfer. Note that clearing the FIFO during a transfer will result in the transfer being aborted.
The ST field starts a new BSC transfer. This has a one shot action, and so the bit will always read back as 0 .
The INTD field enables interrupts at the end of a transfer the DONE condition. The interrupt remains active until the DONE condition is cleared by writing a 1 to the I2CS.DONE field. Writing a 0 to the INTD field disables interrupts on DONE.
The INTT field enables interrupts whenever the FIFO is or more empty and needs writing (i.e. during a write transfer) - the TXW condition. The interrupt remains active until the TXW condition is cleared by writing sufficient data to the FIFO to complete the transfer. Writing a 0 to the INTT field disables interrupts on TXW.
The INTR field enables interrupts whenever the FIFO is or more full and needs reading (i.e. during a read transfer) - the RXR condition. The interrupt remains active until the RXW condition is cleared by reading sufficient data from the RX FIFO. Writing a 0 to the INTR field disables interrupts on RXR.
The I2CEN field enables BSC operations. If this bit is 0 then transfers will not be performed. All register accesses are still permitted however.

Bit(s)	Field Name	Description	Type	Reset
31:16		***Reserved*** *- Write as 0, read as don't care*		
15	I2CEN	I2C Enable 0 = BSC controller is disabled 1 = BSC controller is enabled	RW	0x0
14:11		***Reserved*** *- Write as 0, read as don't care*		
10	INTR	INTR Interrupt on RX 0 = Don t generate interrupts on RXR condition. 1 = Generate interrupt while RXR = 1.	RW	0x0
9	INTT	INTT Interrupt on TX 0 = Don t generate interrupts on TXW condition. 1 = Generate interrupt while TXW = 1.	RW	0x0

8	INTD	INTD Interrupt on DONE 0 = Don t generate interrupts on DONE condition. 1 = Generate interrupt while DONE = 1.	RW	0x0
7	ST	ST Start Transfer 0 = No action. 1 = Start a new transfer. One shot operation. Read back as 0.	RW	0x0
6		***Reserved*** *- Write as 0, read as don't care*		
5:4	CLEAR	CLEAR FIFO Clear 00 = No action. x1 = Clear FIFO. One shot operation. 1x = Clear FIFO. One shot operation. If CLEAR and ST are both set in the same operation, the FIFO is cleared before the new frame is started. Read back as 0. Note: 2 bits are used to maintain compatibility to previous version.	RW	0x0
3:1		***Reserved*** *- Write as 0, read as don't care*		
0	READ	READ Read Transfer 0 = Write Packet Transfer. 1 = Read Packet Transfer.	RW	0x0

S Register

Synopsis The status register is used to record activity status, errors and interrupt requests.
The TA field indicates the activity status of the BSC controller. This read-only field returns a 1 when the controller is in the middle of a transfer and a 0 when idle.
The DONE field is set when the transfer completes. The DONE condition can be used with I2CC.INTD to generate an interrupt on transfer completion. The DONE field is reset by writing a 1 , writing a 0 to the field has no effect.
The read-only TXW bit is set during a write transfer and the FIFO is less than full and needs writing. Writing sufficient data (i.e. enough data to either fill the FIFO more than full or complete the transfer) to the FIFO will clear the field. When the I2CC.INTT control bit is set, the TXW condition can be used to generate an interrupt to write more data to the FIFO to complete the current transfer. If the I2C controller runs out of data to send, it will wait for more data to be written into the FIFO.
The read-only RXR field is set during a read transfer and the FIFO is or more full and needs reading. Reading sufficient data to bring the depth below will clear the field. When I2CC.INTR control bit is set, the RXR condition can be used to generate an interrupt to read data from the FIFO before it becomes full. In the event that the FIFO does become full, all I2C operations will stall until data is removed from the FIFO.
The read-only TXD field is set when the FIFO has space for at least one byte of data. TXD is clear when the FIFO is full. The TXD field can be used to check that the FIFO can accept data before any is written. Any writes to a full TX FIFO will be ignored.
The read-only RXD field is set when the FIFO contains at least one byte of data. RXD is cleared when the FIFO becomes empty. The RXD field can be used to check that the FIFO contains data before reading. Reading from an empty FIFO will return invalid data.
The read-only TXE field is set when the FIFO is empty. No further data will be transmitted until more data is written to the FIFO.
The read-only RXF field is set when the FIFO is full. No more clocks will be generated until space is available in the FIFO to receive more data.
The ERR field is set when the slave fails to acknowledge either its address or a data byte written to it. The ERR field is reset by writing a 1 , writing a 0 to the field has no effect.
The CLKT field is set when the slave holds the SCL signal high for too long (clock stretching). The CLKT field is reset by writing a 1 , writing a 0 to the field has no effect.

Bit(s)	Field Name	Description	Type	Reset
31:10		***Reserved*** *- Write as 0, read as don't care*		
9	CLKT	CLKT Clock Stretch Timeout 0 = No errors detected. 1 = Slave has held the SCL signal low (clock stretching) for longer and that specified in the I2CCLKT register Cleared by writing 1 to the field.	RW	0x0
8	ERR	ERR ACK Error 0 = No errors detected. 1 = Slave has not acknowledged its address. Cleared by writing 1 to the field.	RW	0x0
7	RXF	RXF - FIFO Full 0 = FIFO is not full. 1 = FIFO is full. If a read is underway, no further serial data will be received until data is read from FIFO.	RO	0x0

6	TXE	TXE - FIFO Empty 0 = FIFO is not empty. 1 = FIFO is empty. If a write is underway, no further serial data can be transmitted until data is written to the FIFO.	RO	0x1
5	RXD	RXD - FIFO contains Data 0 = FIFO is empty. 1 = FIFO contains at least 1 byte. Cleared by reading sufficient data from FIFO.	RO	0x0
4	TXD	TXD - FIFO can accept Data 0 = FIFO is full. The FIFO cannot accept more data. 1 = FIFO has space for at least 1 byte.	RO	0x1
3	RXR	RXR - FIFO needs Reading (full) 0 = FIFO is less than full and a read is underway. 1 = FIFO is or more full and a read is underway. Cleared by reading sufficient data from the FIFO.	RO	0x0
2	TXW	TXW - FIFO needs Writing (full) 0 = FIFO is at least full and a write is underway (or sufficient data to send). 1 = FIFO is less then full and a write is underway. Cleared by writing sufficient data to the FIFO.	RO	0x0
1	DONE	DONE Transfer Done 0 = Transfer not completed. 1 = Transfer complete. Cleared by writing 1 to the field.	RW	0x0
0	TA	TA Transfer Active 0 = Transfer not active. 1 = Transfer active.	RO	0x0

DLEN Register

Synopsis The data length register defines the number of bytes of data to transmit or receive in the I2C transfer. Reading the register gives the number of bytes remaining in the current transfer.
The DLEN field specifies the number of bytes to be transmitted/received. Reading the DLEN field when a transfer is in progress (TA = 1) returns the number of bytes still to be transmitted or received. Reading the DLEN field when the transfer has just completed (DONE = 1) returns zero as there are no more bytes to transmit or receive. Finally, reading the DLEN field when TA = 0 and DONE = 0 returns the last value written. The DLEN field can be left over multiple transfers.

Bit(s)	Field Name	Description	Type	Reset
31:16		***Reserved*** *- Write as 0, read as don't care*		

15:0	DLEN	Data Length. Writing to DLEN specifies the number of bytes to be transmitted/received. Reading from DLEN when TA = 1 or DONE = 1, returns the number of bytes still to be transmitted or received. Reading from DLEN when TA = 0 and DONE = 0, returns the last DLEN value written. DLEN can be left over multiple packets.	RW	0x0

A Register

Synopsis The slave address register specifies the slave address and cycle type. The address register can be left across multiple transfers
The ADDR field specifies the slave address of the I2C device.

Bit(s)	Field Name	Description	Type	Reset
31:7		***Reserved*** *- Write as 0, read as don't care*		
6:0	ADDR	Slave Address.	RW	0x0

FIFO Register

Synopsis The Data FIFO register is used to access the FIFO. Write cycles to this address place data in the 16-byte FIFO, ready to transmit on the BSC bus. Read cycles access data received from the bus.
Data writes to a full FIFO will be ignored and data reads from an empty FIFO will result in invalid data. The FIFO can be cleared using the I2CC.CLEAR field.
The DATA field specifies the data to be transmitted or received.

Bit(s)	Field Name	Description	Type	Reset
31:8		***Reserved*** *- Write as 0, read as don't care*		
7:0	DATA	Writes to the register write transmit data to the FIFO. Reads from register reads received data from the FIFO.	RW	0x0

DIV Register

Synopsis The clock divider register is used to define the clock speed of the BSC peripheral. The CDIV field specifies the core clock divider used by the BSC.

Bit(s)	Field Name	Description	Type	Reset
31:16		***Reserved*** *- Write as 0, read as don't care*		
15:0	CDIV	Clock Divider SCL = core clock / CDIV Where core_clk is nominally 150 MHz. If CDIV is set to 0, the divisor is 32768. CDIV is always rounded down to an even number. The default value should result in a 100 kHz I2C clock frequency.	RW	0x5dc

DEL Register

Synopsis The data delay register provides fine control over the sampling/launch point of the data.
The REDL field specifies the number core clocks to wait after the rising edge before sampling the incoming data.
The FEDL field specifies the number core clocks to wait after the falling edge before outputting the next data bit.
Note: Care must be taken in choosing values for FEDL and REDL as it is possible to cause the BSC master to malfunction by setting values of CDIV/2 or greater. Therefore the delay values should always be set to less than CDIV/2.

Bit(s)	Field Name	Description	Type	Reset
31:16	FEDL	FEDL Falling Edge Delay Number of core clock cycles to wait after the falling edge of SCL before outputting next bit of data.	RW	0x30
15:0	REDL	REDL Rising Edge Delay Number of core clock cycles to wait after the rising edge of SCL before reading the next bit of data.	RW	0x30

CLKT Register

Synopsis The clock stretch timeout register provides a timeout on how long the master waits for the slave to stretch the clock before deciding that the slave has hung.
The TOUT field specifies the number I2C SCL clocks to wait after releasing SCL high and finding that the SCL is still low before deciding that the slave is not responding and moving the I2C machine forward. When a timeout occurs, the I2CS.CLKT bit is set.
Writing 0x0 to TOUT will result in the Clock Stretch Timeout being disabled.

Bit(s)	Field Name	Description	Type	Reset
31:16		***Reserved*** *- Write as 0, read as don't care*		
15:0	TOUT	TOUT Clock Stretch Timeout Value Number of SCL clock cycles to wait after the rising edge of SCL before deciding that the slave is not responding.	RW	0x40

3.3 10 Bit Addressing

10 Bit addressing is an extension to the standard 7-bit addressing mode. This section describes in detail how to read/write using 10-bit addressing with this I2C controller.

10-bit addressing is compatible with, and can be combined with, 7 bit addressing. Using 10 bits for addressing exploits the reserved combination 1111 0xx for the first byte following a START (S) or REPEATED START (Sr) condition.

The 10 bit slave address is formed from the first two bytes following a S or Sr condition.

The first seven bits of the first byte are the combination 11110XX of which the last two bits (XX) are the two *most significant* bits of the 10-bit address. The eighth bit of the first byte is the R/W bit. If the R/W bit is '0' (write) then the following byte contains the remaining 8 bits of the 10-bit address. If the R/W bit is '1' then the next byte contains data transmitted from the slave to the master.

Writing

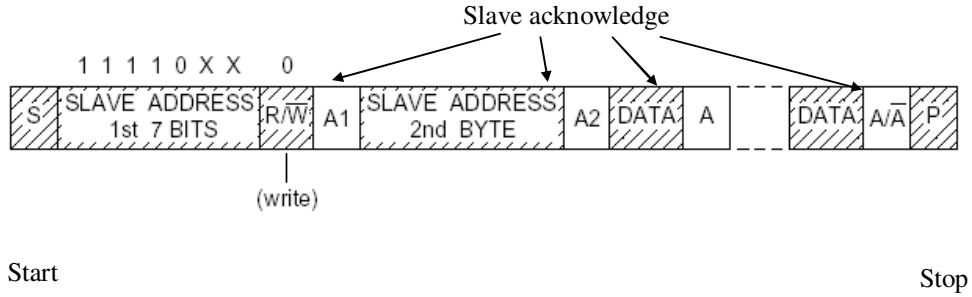

Figure 3-1 Write to a slave with 10 bit address

Figure 3-1 shows a write to a slave with a 10-bit address, to perform this using the controller one must do the following:

Assuming we are in the 'stop' state: (and the FIFO is empty)

1. Write the number of data bytes to written (plus one) to the I2CDLEN register.
2. Write 'XXXXXXXX' to the FIFO where 'XXXXXXXX' are the least 8 significant bits of the 10-bit slave address.
3. Write other data to be transmitted to the FIFO.
4. Write '11110XX' to Slave Address Register where 'XX' are the two most significant bits of the 10-bit address. Set I2CC.READ = 0 and I2CC.ST = 1, this will start a write transfer.

Reading

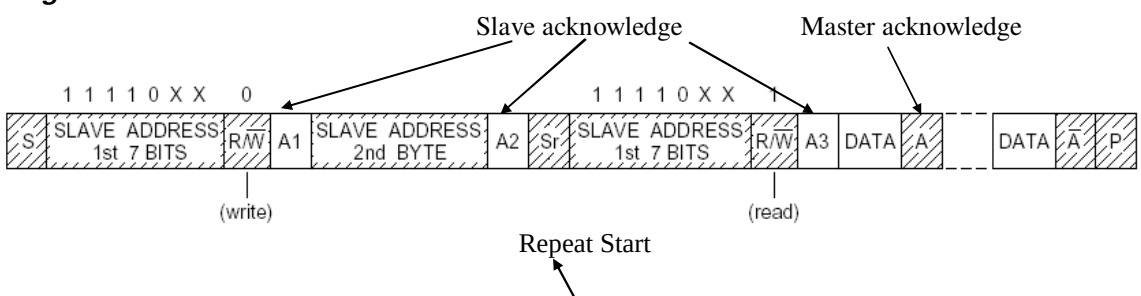

Figure 3-2 Read from slave with 10 bit address

Figure 3-2 shows how a read from a slave with a 10-bit address is performed. Following is the procedure for performing a read using the controller:

1. Write 1 to the I2CDLEN register.
2. Write 'XXXXXXXX' to the FIFO where 'XXXXXXXX' are the least 8 significant bits of the 10-bit slave address.
3. Write '11110XX' to the Slave Address Register where 'XX' are the two most significant bits of the 10-bit address. Set I2CC.READ = 0 and I2CC.ST = 1, this will start a write transfer.
4. Poll the I2CS.TA bit, waiting for the transfer has started.
5. Write the number of data bytes to read to the I2CDLEN register.
6. Set I2CC.READ = 1 and I2CC.ST = 1, this will send the repeat start bit, new slave address and R/W bit (which is '1') initiating the read.