Application Report

Lit. Number – December 2009

Proposal for I2C Host Interface Implementation for BSL

Kasthuri Annamalai *MSP430 Applications*

ABSTRACT

This application report describes the I2C Interface Implementation for the BSL. The document details the architecture of the I2C host and the communication protocol with the BSL slave.

1 I2C Protocol

1.1 Basic Protocol with Byte Level Acknowledge

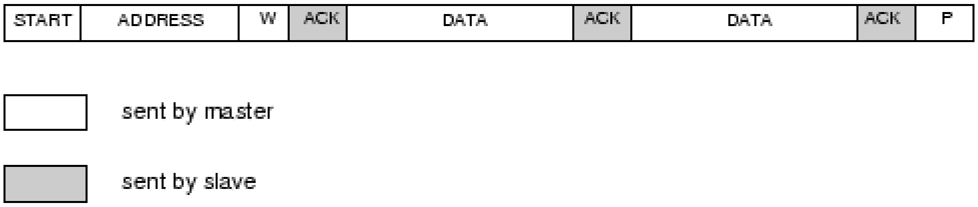

Figure 1. Basic Protocol - Byte Level ACK

1. Send the START bit
2. Send the slave address
3. Send the Read(R)-1 / Write (W)-0 bit.
4. Wait for/Send an acknowledge bit
5. Send/Receive the data byte (8 bits)
6. Expect/Send acknowledge bit
7. Send the STOP bit

1.2 I2C Protocol for BSL - Read from Slave

1. Send a start sequence (S)
2. Send I2C address of Slave with the R/W bit low (even address) (ADDR) + W.
3. Send Data or address of internal address of slave register (NDATA)
4. Send a start sequence again (repeated start) (Res)
5. Send I2C address of slave with the R/W bit high (odd address) (ADDR)
6. Read data byte from slave (RDATA)
7. Send the stop sequence (P)

All BSL wrapper commands from the host/master are considered as data (represented as NDATA, they may consist of n number of bytes) and all data read from the slave are also considered data and represented as RDATA.

The protocol we intend to use for all communication from Master to Slave is as follows:

From Master -> S + ADDR + W + NDATA + ReS + ADDR + R

From Slave -> RDATA

From Master -> P

1.3 ACKNOWLEDGE – ACK

There are 2 levels of acknowledge we are concerned about.

- The low level acknowledges indicating reception of each byte that is part of the I2C protocol. This will be handled by the hardware if proper I2c settings are set on the slave registers.

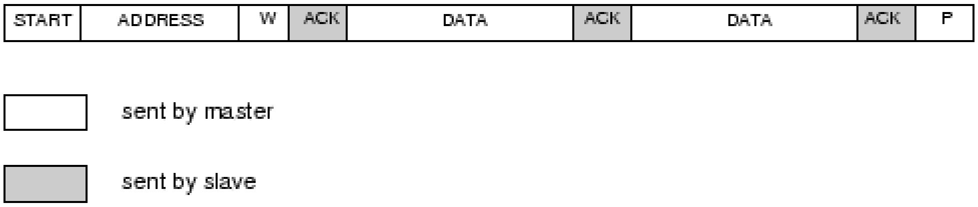

Figure 2. Byte Level ACK

- The higher level acknowledge indicating that the checksum of the BSL core command obtained is correct and as expected and in some cases may indicate the command was properly executed. This is the first byte of RDATA. If this is NAK (other than 0x00) then it means proper command was not received and MASTER should consider that command transmission was a failure. If this is 0x00 (ACK) it indicates that the transmission/reception of command was correct with the right checksum and what follows is the response if any from the slave. The slave may keep the CLK line low if it needs a long time to process before it responds to the command.

RDATA

Byte 0	Byte 1…r
ACK/NAK – checksum right/wrong or command execution right/wrong	Data from Slave to Host if any

Figure 3. Checksum ACK