iP-MB100

BiSS INTERFACE MASTER IP

preliminary

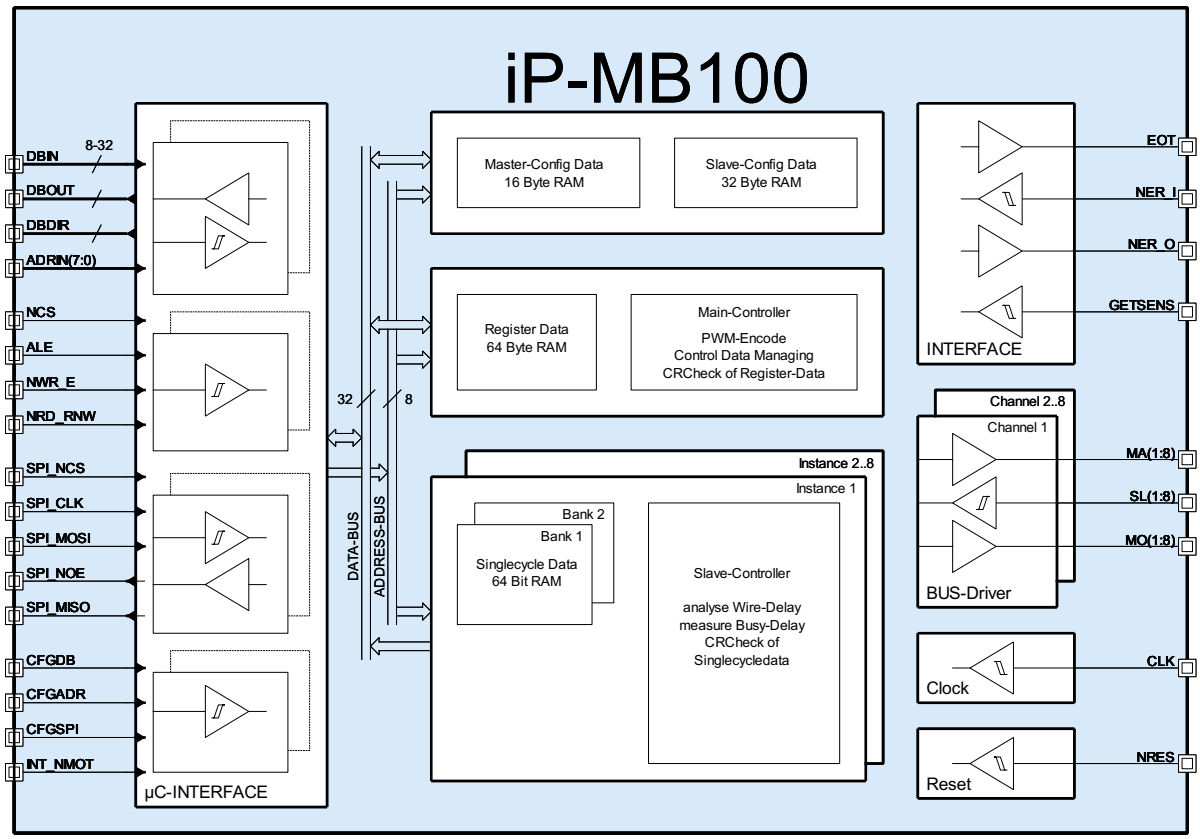
FEATURES

- Bidirectional BiSS communication with up to 8 slaves
- Supports SSI protocol for unidirectional data transmission
- Synchronous sensor data acquisition with cyclic transfer at data rates of up to 10 Mbit/s
- Slave register operations during cyclic data transfers
- Automatic compensation of line delays and conversion times
- Data lengths of up to 64 bit for sensor data, configurable for each slave
- Data verification by CRC polynomials of up to 16 bits per slave
- Separate memory banks enable free controller access during BiSS sensor data transfers
- 64 bytes memory for bidirectional register communication
- Serial controller communication by SPI™-compatible mode
- Parallel interface with 8 bit data/address bus for intel- and Motorola-devices with combined data and address bus
- Parallel interface with 8 /16 32 bit data/address bus for arm devices (AXI)

APPLICATIONS

- Bidirectional communication in multi sensor systems
- Linear and rotary encoders
- Motor feedback systems
- PLC systems
- Drives

BLOCK DIAGRAM



DESCRIPTION

iP-MB100 is a single-chip BiSS/SSI interface master controller featuring an 8-bit bus interface to industrial standard microcontroller. Alternatively an SPI interface enables serial communication between iP-MB100 and the connected microcontroller. The BiSS devices including one or several sensors are connected to the clock line MA and the data return line SL via point-to-point connection (Figure 1). The BiSS devices including actuators and sensors are additionally connected to the data line MO. It is possible to connect multiple devices in chain (Figure 2). A maximum of eight BiSS slaves are supported, each with it's own configurable data sections covering:

1. Sensor / actuator data from 0 to 64 bits
   (for measurement data, flags like alarm and warning, Sign of Life cycle counter, . . .)
2. Register data with 64 directly addressable bytes and 256 banks with 64 bytes per slave ID
   (e.g. for device parameters)

iP-MB100 provides two RAM memory banks for each slave enabling simultaneous access by the microcontroller while new sensor data is being read. A 64-byte memory supports register transfers. Sensor data acquisition is started by a microcontroller command or via signal GETSENS. Alternatively, iP-MB100 can also read new sensor data automatically. In this case the cycle time can be set as required. The end of the sensor data acquisition and transmission is indicated at the signal EOT. In case of an error during transmission the signal NER signals a digital low. Errors during communication can be verified by the microcontroller via a status register. A system error message can also access this register if the input message signal NER_I is kept digital low by an external intervention. iP-MB100 generates a clock signal for sensor communication using an external 20 MHz clock signal. iP-MB100 is based on the BiSS master IP family MB100_X.

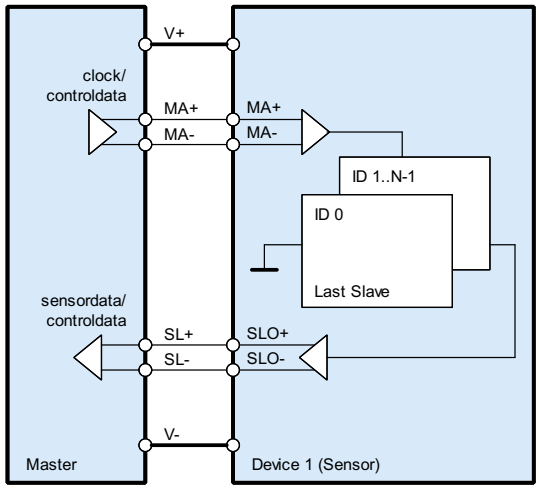


Figure 1: Point-to-point connection of iP-MB100 to one device with several slaves

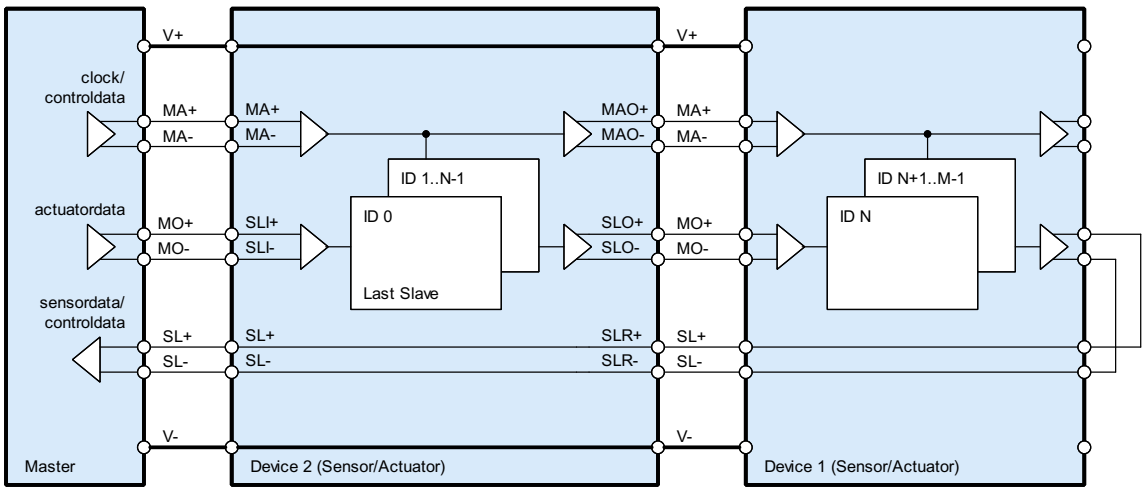


Figure 2: Example network of iP-MB100 and two devices

The IP offered here is a multifunctional IP that contains integrated BiSS C interface components. The BiSS C process is protected by patent DE 10310622 B4 owned by iC-Haus GmbH. Users benefit from the open BiSS C protocol with a free license which is necessary when using the BiSS C protocol in conjunction with this IP.
Contact for the license at www.biss-interface.com/

ELECTRICAL CHARACTERISTICS

Item No.	Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
I/O Ports: EOT, NER, GETSENS							
C01	Tw(GETSENS)	Minimum duration of GETSENS pulse		1/f(CLK)			
Oscillator: CLK							
D01	f(CLK)	Permissible external Clock Rate at CLK	Duty cycle 48% . . . 52%		20	25	MHz
Reset Signal Input: NRES							
E01	td()res	Required Reset Pulse Duration	At NRES	250			ns

OPERATING REQUIREMENTS: BiSS Interface - BiSS B/C frame

Operating conditions: register bit SELSSI = 0; Alias: MA = MA1/MA2_NMA1, SL = SL1/SL2_NSL1

Item No.	Symbol	Parameter	Conditions	Min.	Max.	Unit
Frame						
I001	TMAS	Clock Period	FreqSens via FREQ(4:0) selected in accordance with Table 52 on page 25	2	320	1/f(CLK)
I002	tMASl	Clock Signal Lo Level Duration		50	50	% TMAS
I003	tMASh	Clock Signal Hi Level Duration		50	50	% TMAS
I004	tpLine	Permissible Line Delay		0	indefinite	
I005	Δ tpL	Permissible Propagation Delay of Subsequent Clock Cycles vs. 1st Clock Cycle	Δ tpL = max(|tpLine - tpLx|); x= 1 ... n		25	% TMAS
I006	Ttos	Permissible Timeout (Slave)		55		% TMAS

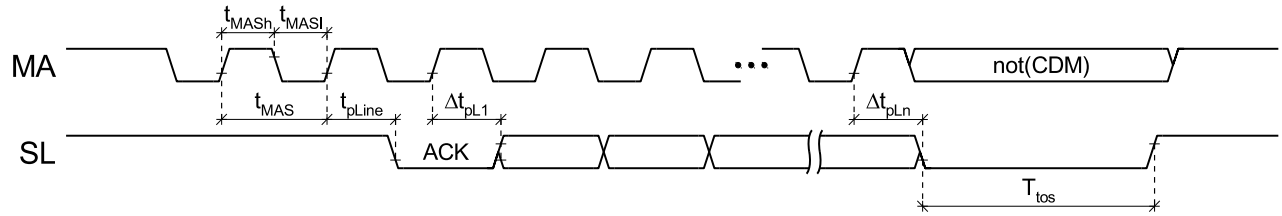


Figure 3: Timing diagram BiSS B/C frame

SLx line sampling
Line delays longer than one clock cycle are permissible in BiSS.
Within one MA clock cycle four equally distributed sampling points are used to evaluate the overall delay from the second rising edge at MAx to the first falling edge at SLx (ACK).
Refer also to the characteristics in BiSS Interface PROTOCOL DESCRIPTION.

OPERATING REQUIREMENTS: BiSS Interface - BiSS B register data cycle

Operating conditions: register bit SELSSI = 0; Alias: MA = MA1/MA2_NMA1, SL = SL1/SL2_NSL1

Item No.	Symbol	Parameter	Conditions	Min.	Max.	Unit
I101	TMAR	Clock period	FreqReg via FREQ(7:5) selected in accordance with Table 53 on page 25	2	256	TMAS
I102	tMA0h	”Logic 0” hi Level duration		25	25	% TMAR
I103	tMA1h	”Logic 1” hi level duration		75	75	% TMAR
I104	tMAth	Clock Signal hi Level duration	register data readout	50	50	% TMAR
I105	tsSM	Setup Time: SL stable before MA lo→hi		30		ns
I106	thSM	Hold Time: SL stable after MA lo→hi		0		ns
I107	Ttor	Permissible timeout (slave)	For Ttos details see item i306 BiSS Frame	80		% TMAR

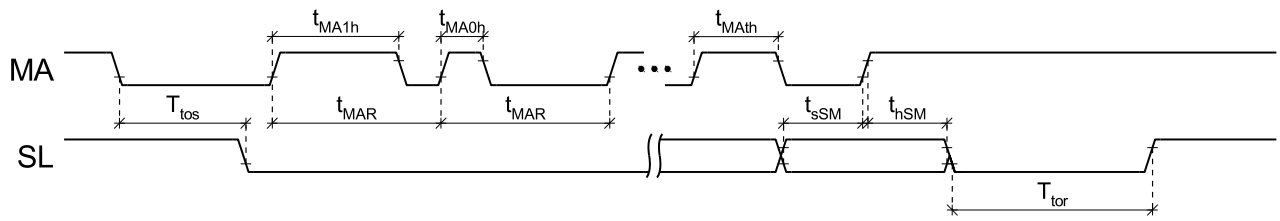


Figure 4: Timing diagram BiSS B register access

OPERATING REQUIREMENTS: BiSS Interface (SSI mode)

Operating conditions: register bit SELSSI = 1; Alias: MA = MA1/MA2_NMA1, SL = SL1/SL2_NSL1

Item No.	Symbol	Parameter	Conditions	Min.	Max.	Unit
I201	TMAS	Clock period	FreqSens via FREQ(4:0) selected in accordance with Table 52 on page 25	2	320	1/f(CLK)
I202	tMASh	Clock Signal Hi level duration		50	50	% TMAS
I203	tMASl	Clock signal Lo level duration		50	50	% TMAS
I204	tsSM	Setup Time: SLx stable before MAx lo→hi		30		ns
I205	thSM	Hold Time: SLx stable after MAx lo→hi		0		ns
I206	Ttos	Permissible timeout (Slave)		55		% TMAS

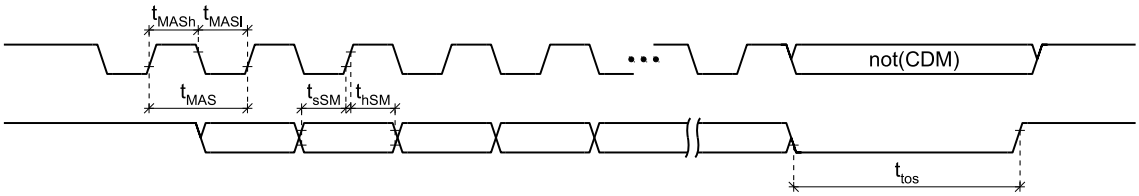


Figure 5: Timing diagram SSI mode

SLx line sampling

In SSI interface mode SLx values are sampled with the rising edge at MAx. An overall delay of the sensor response to the clock at MAx, caused by process times in the sensor or transmission times, is permissible up to the length of one clock cycle minus tsSM.

IP FILE LIST

Directory **source:**

biss_sim.vhd	includes package biss_sim for simulation
biss_synth.vhd	includes package biss_synth for synthesis
mb.package.vhd	Package with constants, types and functions
mb_vcomponents.vhd	component declaration of mb*
mbz101.vhd	Control Communication
mb1.vhd	State machine for master
mbz201.vhd	CRC calculation for single cycle data
mb2.vhd	State machine for Single cycle data
mbz402.vhd	SPI interface
mb4.vhd	µC interface
mbz501.vhd	Input sample
mb5.vhd	Channel select logic
mb6.vhd	Ram for control communication
mb7.vhd	Configuration RAM
mb8.vhd	Configuration Single cycle data
mb9.vhd	RAM for Singe cycle data
mc200*.vhd	Simulation model for BiSS-C slaves
ram.entity.vhd	RAM Entity
ram.syn.vhd	RAM RTL architecture
ram.cyclone.vhd	RAM architecture for use with Intel/Altera Quartus™
mb101xe.vhd	Example implementation

Directory **validation:**

mb101x_tb	Test bench
mb101x.vsim.tcl	Control file for simulation with modelsim™

Directory **implementation:**

mb101x.cycloneii.quartus	Example implementation for use with Intel/Altera Quartus™

IP GENERICS

The GENERICS of the BiSS master IP MB100 are used for a static configuration of the design and to control the resource usage during the synthesis flow.

Revision	
0	First release
1... 255	Increment by one with each revision

Table 5: Revision number

Version	
64	Device identifier MB100
65... 127	MB101...MB163

Table 6: Device identifier

Note: VERSION and REVISION are provided by the IP and may not be changed individually.

BusWidth, RamWidth	
8, 8	Data bus width 8 bit (DB7...DB0)
16, 16	Data bus width 16 bit (DB15...DB0)
16, 8	CFGDB = 0: Data bus width 16 bit CFGDB = 1: Data bus width 8 bit
32, 32	Data bus width 32 bit (DB31...DB0)
32, 16	CFGDB = 0: Data bus width 32 bit CFGDB = 1: Data bus width 16 bit
32, 8	CFGDB = 0: Data bus width 32 bit CFGDB = 1 and DB16 = 0: Data bus width 16 bit CFGDB = 1 and DB16 = 1: Data bus width 8 bit

Table 7: Bus- and RAM- width of data bus

AddressBus	
0	Use address latch No address ports available
1	Address latch and address ports enabled CFGADR = 0: Use address ports CFGADR = 1: Use address latch
2	Use address ports No address latch available

Table 8: Address latch implementation

UseSyncIf	
0	Do not use synchronous interface
1	Use synchronous interface

Table 9: Interface implementation

UseSPI	
0	Do not use SPI interface
1	Use SPI interface (requires RamWidth = 8)

Table 10: SPI implementation

Slaves	
1... 8	Slave count 64 bit single cycle data RAM for sensor resp. actuator data (Addr. 0x00...0x7F) and 16 bit configuration RAM (Addr. 0xC0...0xDF)

Table 11: Slave count

Channels	
1... 8	BiSS channel count

Table 12: Channel count

NumRegs	
1... 64	Byte count for register communication

Table 13: Register count

BankCount	
1	Use one bank for single cycle data
2	Use two banks for single cycle data

Table 14: Number of switchable banks

ClkdivSensLen	
0	No clock counter for sensor data frequency
1... 4	Factor: $2^1...2^{ClkdivSensLen+1}$
5	Factor: (1 or 10) $*$ $2^1...2^5$

Table 15: Pre clock divider for FREQS

ClkdivAgsLen	
0	No clock counter for a fixed cycle time at automatic data request
1... 7	Factor: 20 $*$ $2^1...2^{ClkdivAgsLen}$
8	Factor: (20 or 625) $*$ $2^1...2^7$

Table 16: Pre clock divider for FREQAGS

ClkdivRegLen	
0	No clock counter for register data frequency
1... 7	Factor: $2^1...2^{ClkdivRegLen+1}$

Table 17: Pre clock divider for FREQAGS

BusyCountLen	
0	No counter for register data process time supervision
1... 12	Counter depth: $2^{BusyCountLen}$

Table 18: Busy counter depth for register access

BusySCountLen	
0	No counter for start bit delay
1... 8	Counter depth: $2^{BusySCountLen}$

Table 19: Clock counter for start bit delay

UseReg245	
0	No test register at address 245
1	Test register at address 245 enabled

Table 20: Use of test register 0xF5 (245) use

IP PORTS

Name	Direction	Width	Function
CLK	IN	1	External Clock Input
NRES	IN	1	Asynchronous Reset Input, low active
Configuration			
CFGADR	IN	1	Address Bus Select Input (1 = external bus, 0 = Internal latch)
CFGDB	IN	1	Data Bus Width Select Input (1 = 8/16 bit, 0 = 32 bit)
CFGSPI	IN	1	Serial/Parallel Mode Select Input (1 = Serial, 0 = Parallel)
INT_NMOT	IN	1	Mode Select (1 = Intel, 0 = Motorola)
Parallel host interface			
NCS	IN	1	Chip Select Input, low active
ADRIN	IN	8	Address Bus (used if CFGADR = 1)
ALE	IN	1	Address Latch Enable Input (used if CFGADR = 0)
NWR_E	IN	1	Write Enable Input, low active (Intel) Enable Input, high active (Motorola)
NRD_RNW	IN	1	Read Enable Input, low active (Intel) Read/Not-Write Select Input (Motorola)
DBIN	IN	Bus width	Data Bus Input
DBOUT	OUT	Bus width	Data Bus Output
DBDIR	OUT	1	Data Bus Direction (1 = Input, 0 = Output)
Serial host interface			
SPI_NCS	IN	1	SPI Chip Enable, low active
SPI_CLK	IN	1	SPI Clock
SPI_NOE	OUT	1	SPI Output Enable, low active
SPI_MOSI	IN	1	SPI Serial Input
SPI_MISO	OUT	1	SPI Serial Output
BiSS C Signals (connect to external RS422 transceiver)			
MA	OUT	Channels	BiSS Clock Line Output
MO	OUT	Channels	BiSS Data Line Output
SL	IN	Channels	BiSS Data Line Input
Control Signals			
GETSENS	IN	1	Sensor Data Request
EOT	OUT	1	End-Of-Transmission
NER_I	IN	1	Error input (low active)
NER_O	OUT	1	Error output (low active)

The ports NER_I and NER_O could be combined to build a open drain port.

CONFIGURATION PARAMETERS

REGISTER MAP

OVERVIEW								
Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sensor and Actuator Data								
0x00	SCDATA1(7:0)							
0x01 .. 0x07	SCDATA1(63:8)							
0x08 .. 0x3F	SCDATA2(63:0) .. SCDATA8(63:0)							
0x40 .. 0x7F	– *							
Register Data								
0x80 †	RDATA1(7:0)							
0x80 ‡	IDS(7:0)							
0x81 .. 0xBF	RDATA2(7:0) .. RDATA64(7:0)							
Configuration Slave 1								
0xC0	GRAYS1 / LSTOP1	ENSCD1	SCDLEN1(5:0)					
0xC1	SELCRCS1	SCRCLEN1(6:0) / SCRCPOLY1(7:1)						
0xC2	SCRCSTART1(7:0)							
0xC3	SCRCSTART1(15:8)							
0xC4 .. 0xDF	Configuration Slave 2(31:0) .. Configuration Slave 8(31:0)							
Control Communication Configuration								
0xE0	– *							
0xE1	– *							
0xE2	WNR	REGADR(6:0)						
0xE3	– *		REGNUM(5:0)					
0xE4	CHSEL(8:1)							
0xE5 †	CTS	REGVERS	SLAVEID(2:0)			– *	EN_MO	HOLDCDM
0xE5 ‡	CTS	REGVERS	CMD(1:0)		IDA_TEST	– *	EN_MO	HOLDCDM
Master Configuration								
0xE6	FREQR(2:0)			FREQS(4:0)				
0xE7	– *						NOCRC	SINGLEBANK
0xE8	FREQAGS(7:0)							
0xE9	MO_BUSY(7:0)							
0xEA	REVISION(7:0)§							
0xEB	VERSION(7:0)§							

* Reserved or unused register bits highlighted as '–' need to be written with 0 if a byte wide register write access is required.
† Using register access in control communication.
‡ Using command/instructions in control communication.
§ Register bits with constant '0' or '1' are ROM-based values and can not be changed through writing.

OVERVIEW								
Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Channel Configuration								
0xEC	SLAVELOC8	SLAVELOC7	SLAVELOC6	SLAVELOC5	SLAVELOC4	SLAVELOC3	SLAVELOC2	'1'
0xED	CFGCH4(1:0)		CFGCH3(1:0)		CFGCH2(1:0)		CFGCH1(1:0)	
0xEE	CFGCH8(1:0)		CFGCH7(1:0)		CFGCH6(1:0)		CFGCH5(1:0)	
Slave Configuration 2								
0xEF	ACTnSENS(8:1)							
Status Information								
0xF0	nERR	nAGSERR	nDELAYERR	nSCDERR	nREGERR	REGEND	'1'	EOT
0xF1	SVALID4	'0'	SVALID3	'0'	SVALID2	'0'	SVALID1	'0'
0xF2	SVALID8	'0'	SVALID7	'0'	SVALID6	'0'	SVALID5	'0'
0xF3	CDMTIME-OUT	CDSSEL	REGBYTES(5:0)					
Instruction Register								
0xF4	BREAK	HOLDBANK	SWBANK	INIT	INSTR(2:0)			AGS
0xF5	MAVO	MAFO	MAVS	MAFS	– *	– *	ENTEST	CLKENI
0xF6	– *	– *	– *	– *	– *	– *	– * *	– *
0xF7	– *	– *	– *	– *	– *	– *	– *	– *
Status Information 2								
0xF8	CDS4	SL4	CDS3	SL3	CDS2	SL2	CDS1	SL1
0xF9	CDS8	SL8	CDS7	SL7	CDS6	SL6	CDS5	SL5
0xFA	– *	– *	– *	– *	– *	– *	– *	– *
0xFB	– *	– *	– *	– *	– *	– *	– *	SWBANK-FAILS
Reserved								
0xFC .. 0xFF	– *	– *	– *	– *	– *	– *	– *	– *

Table 21: Register layout

Note: Reserved registers need to be written with 0x00.

Note: iP-MB100 does reset all RAM registers to '0' on a power on reset.

FUNCTIONAL DESCRIPTION

The open source BiSS Interface Protocol implements a realtime interface for digital, serial and secure communication between drive, sensor and actuator. In the point-to-point configuration BiSS uses one clock line MAx from the master to the sensor and one data line SLx from the sensor to the master (see Figure 1). A device may contain multiple slaves. The data input signal SLI of the last slave is set to digital low. The slaves are daisy-chained (SLO $\rightarrow$ SLI) and the data output signal SLO of the first slave is directly connected to the master. A data line from the master to the slave is not mandatory.

In the bus configuration the data output line MO is used to transfer actuator data from the master to the slaves (see figure 2). The BiSS protocol describes cyclic data frames and differentiates between process data and control data. Process data are completely transmitted in each frame (SCD) used as actuator or sensor data and control data are transmitted one bit per frame (CD) used for commands and register access.

BiSS C Frame
The BiSS C Frame starts with a digital high on the clock line MAx and is notified by iP-MB100 with a falling edge at the signal EOT. On the first falling edge of MAx all slaves check the SLI signal for a digital low determining the last slave. With the first rising edge at MAx all sensors start calculating their sensor data. The second rising edge of MAx forces all slaves to acknowledge the BiSS-C frame with a falling edge at SLO. The master uses the acknowledge to measure the line delay. When the sensor data calculation is finished, the last slave in the chain generates the start bit which will be passed synchronously through all slaves to the master. If the master data output MOx is used, the start bit delay has to be configured to ensure sufficient processing time regarding sensor data for all slaves. Subsequent to the start bit follows one control data bit for all slaves (CDS) which is set according to the rules of the control frame. After the CDS bit, the process data including sensor and actuator data is sent with the most significant bit (MSB) first. At the end, the master sends it's control data bit (CDM) inverted on the clock line MAx to conclude the BiSS-C Frame.

During processing the frame all slaves observe the MAx clock line and change into the timeout state, if MAx is stable for a specific time defined within each slave. In the timeout state, only the last slave forces it's SLO signal to digital high. The other slaves in the chain connect SLI and SLO to signal the master that all slaves are in the timeout state. After detecting the slave's timeout with SLO = 1, the master may changes the MAx clock line to digital high or keeps the clock line constant until the next frame begins. This is advantageous if the BiSS-C Frame has not been clocked out completely, e.g. for a fast configuration phase and high control data transmission rates. The difference is indicated in Figures 6 and 7.

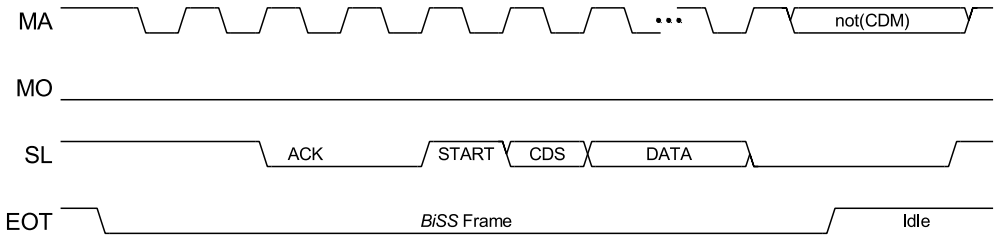


Figure 6: BiSS C frame in point-to-point configuration

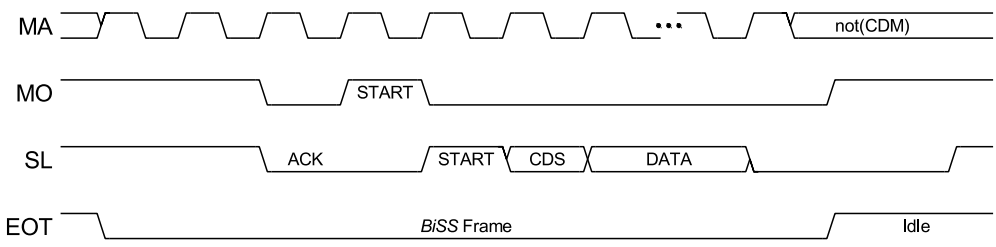


Figure 7: BiSS C frame in bus configuration

BiSS C Initialization Sequence

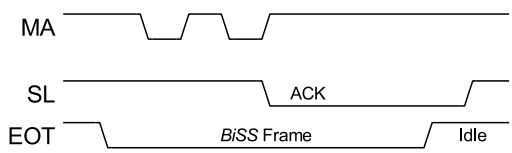


Figure 8: BiSS C initialization sequence

An initialization (INIT) sequence is necessary, if the last slave in the point-to-point configuration is not defined and the data line SLx is digital low. In the initialization sequence, two digital low pulses are generated at MAx. The slave(s) should answer with a falling-edge after the second pulse and with a rising-edge at SLx after the BiSS timeout. The time between the second rising-edge at MAx and the falling-edge at SLx is measured as line delay and stored in the single cycle data RAM, see chapter INSTRUCTION REGISTER on page 30.

Extended SSI BiSS C Register Communication

The extended SSI operation enables the BiSS C register write access to SSI slaves. The master is able to transmit a BiSS C register write access to the slave without the slave's CDS feedback. The master cannot verify that the BiSS C register write access to the slave did succeed. At the end of the SCD frame the master sends the CDM bit inverted on the MA clock line.

BiSS B Register Communication

For the BiSS B protocol the register communication is started by a timing condition and a handshake at the beginning of the frame (see Figure 9). Alternatively, the register communication can be activated at the beginning of the frame with a connected MO line (slave ID "0" remains unused).

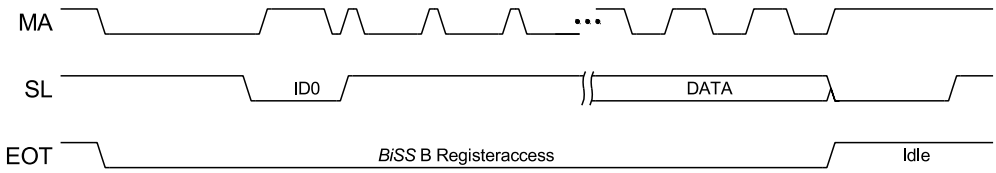


Figure 9: BiSS B register access (EN_MO = 0)

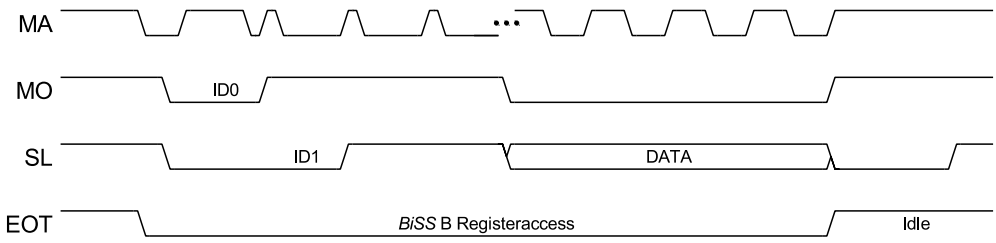


Figure 10: BiSS B register access (EN_MO = 1)

BiSS B Initialization Sequence

In the BiSS-B initialization sequence (init) only the timing handshake with the first falling edge at MA is executed.

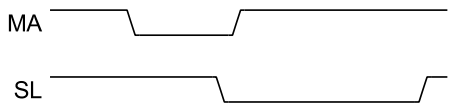


Figure 11: BiSS B initialization sequence

MICROCONTROLLER INTERFACE

The iP-MB100 supports a parallel and a serial microcontroller interface. The input port CFGSPI is used to select the desired interface. The interfaces cannot be used at the same time.

Parallel Microcontroller Interface

With signal CFGSPI = 0 the 8 bit parallel microcontroller interface is selected in which the bidirectional data bus alternately transmits addresses and data in blocks of 8 bits. The signal INT_NMOT selects two different communication types.

CFGSPI	INT_NMOT	Mode
0	0	Motorola 68HC11
0	1	Intel 8051

Table 22: Parallel communication types

In both cases, a digital low at the signal NCS activates the interface and the signal ALE is used to store the address. The data signals are tristate while deactivated via NCS.

If the signal INT_NMOT = 0, the motorola communication type is selected. The signal RNW chooses between read and write access and the signal E executes the access (see Figure 12).

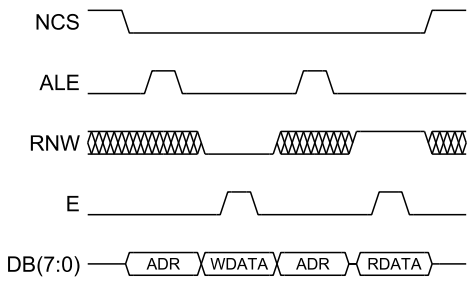


Figure 12: Motorola controller communication

The Intel 8051 controller communication type uses NWR as write enable and NRD as read enable. It is selected with INT_NMOT = 1 (see Figure 13).

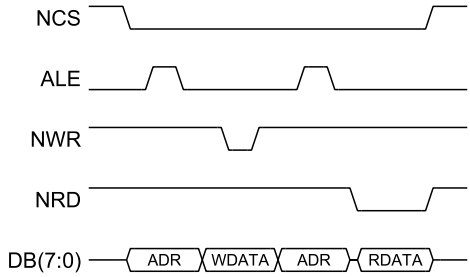


Figure 13: Intel controller communication

SPI Serial Microcontroller Interface

If the signal CFGSPI = 1, the SPI serial microcontroller interfaces with polarity = 0 and phase = 0 are selected.

CFGSPI	INT_NMOT	Mode
1	-	SPI1 (polarity = 0, phase = 0)

Table 23: SPI communication mode

When operating in conjunction with an SPI controller, the signal ALE is used as a clock input (SCK), the signal NCS as an enable input (NCS), the signal DB0 is used as the data input (SI) and the signal DB1 is used as the data output (SO). Data is transmitted serially in successive blocks of 8 bits starting with the command. Depending on the commands, the following blocks represents an address and one or more data bytes (see figure 14).
Six commands are available:

SPI-COMMAND		
Code	Description	Address
0x02	Write Data	Transmitted
0x03	Read Data	Transmitted
0x05	Read Status	0xF0
0x07	Write Instruction	0xF4
0x09	Read Data 0	0x00
0x0B	Write Data 0	0x00

Table 24: Command Codes for SPI

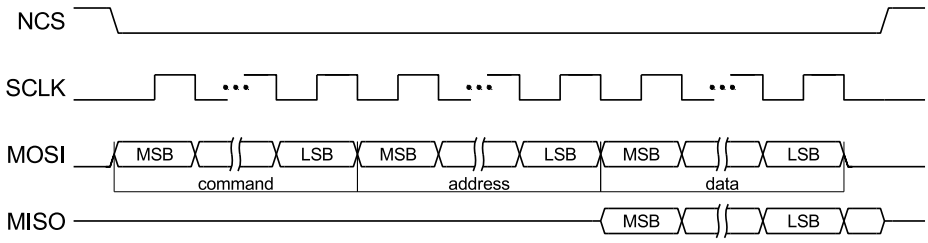


Figure 14: SPI communication

The first two commands in Table 26 can be used to write data to or read data from iP-MB100's registers. In the "read data" command a delay between the address and the first data is necessary (Refer to Op.Require. No. I201). The last four commands are shortened write and read commands with a fixed start address. This means that it is not necessary to send an address as described above. The data can directly be transmitted after the command. With all commands it is possible to transmit several bytes of data consecutively, if the NCS signal remains digital low and ALE/SCK is clocked continuously. The address is internally increased by 1 for each transmitted byte starting at the fixed or transmitted address respectively.

Additional 2nd SPI Serial Microcontroller Interface
If the SPI serial microcontroller interface is activated, an additional SPI interface can be enabled with NWR_E = 0 for dedicated register access with reduced function set.

CFGSPI	NWR_E	Mode
1	0	SPI2, optional (polarity = 0, phase = 0)

Table 25: SPI communication mode

The additional SPI interface at DB4 . . . DB7 is available for an exclusive read access to the SCD single cycle data RAM of the slaves 5 to 8 with the commands "Read Data" and "Read Data 0". Access to the status, the instruction and the parameter registers is not possible.

SPI-COMMAND		
Code	Description	Address
0x03	Read Data	Transmitted
0x09	Read Data 0	0x00

Table 26: Command Codes for SPI2

Note: This 2nd SPI can only be used to read SCDATA from 0x40 . . . 0x7F. It is not possible to read the status or to access the configuration RAM via SPI2.

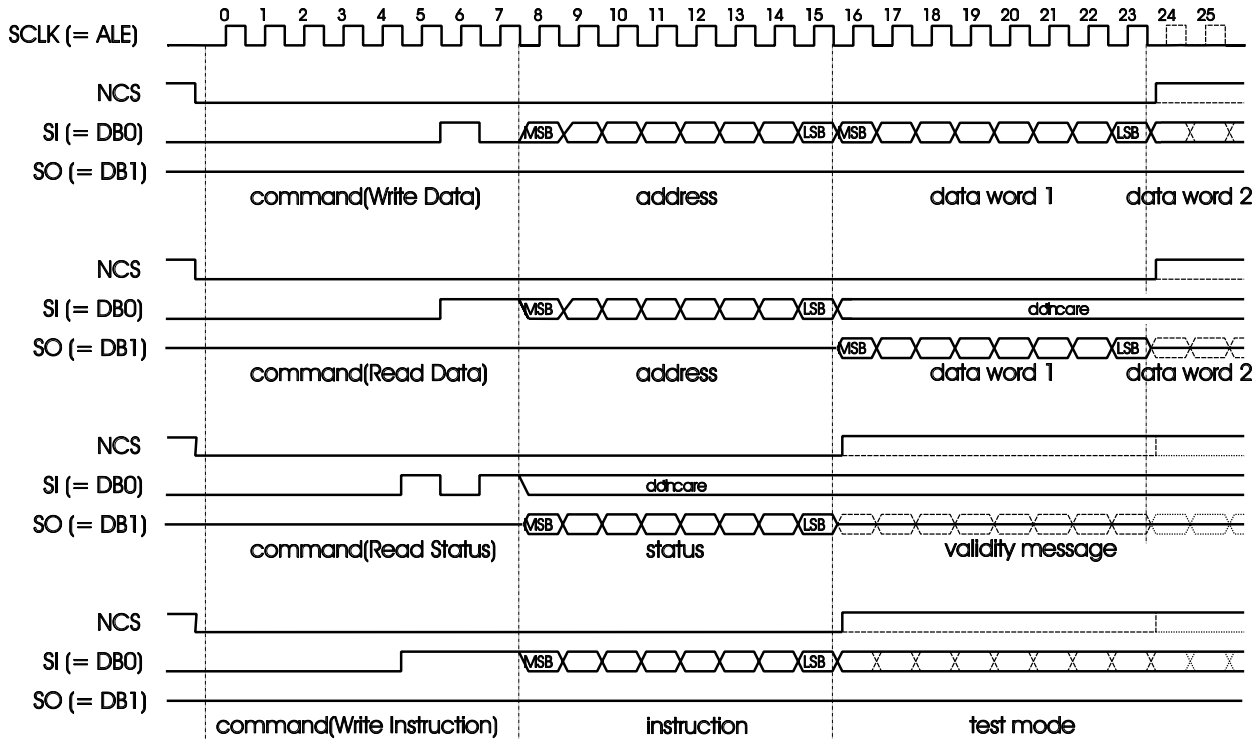


Figure 15: SPI transmission protocol (polarity 0, phase 0)

SENSOR AND ACTUATOR DATA

The process data memory buffers the sensor and actuator data and has eight bytes of memory for each slave. Eight slaves are supported. The address mapping is shown in Table 27.

SCDATA	Addr. 0x00 . . . 0x3F; bit 7:0	R/W
Addr.	Content	
0x00	SCDATA1(7:0)	
0x01	SCDATA1(15:8)	
0x02	SCDATA1(23:16)	
0x03	SCDATA1(31:24)	
0x04	SCDATA1(39:32)	
0x05	SCDATA1(47:40)	
0x06	SCDATA1(55:48)	
0x07	SCDATA1(63:56)	
0x08 . . 0x0F	SCDATA2(63:0)	
0x10 . . 0x17	SCDATA3(63:0)	
0x18 . . 0x1F	SCDATA4(63:0)	
0x20 . . 0x27	SCDATA5(63:0)	
0x28 . . 0x2F	SCDATA6(63:0)	
0x30 . . 0x37	SCDATA7(63:0)	
0x38 . . 0x3F	SCDATA8(63:0)	

Table 27: Address mapping of sensor data

The sensor data is arranged in the memory area with the least significant bit (LSB) at the lowermost address at bit position 0. The memory is written byte-by-byte and unused bits are set to zero. Unused bytes remain unchanged. If there is sufficient free memory available for CRC data, the read and inverted CRC bits are stored beginning at the highest address downwards. The storage of the CRC can be disabled with NOCRC (see Table 55).

Example: BiSS Sensor Bus with 3 Slaves
The following example shows the address mapping of the sensor data and CRC bits generated by three slaves.

Slave 1: 19+2 bits of sensor data, 6 bits of CRC
=> total length of 27 bits

Slave 2: 12+2 bits of sensor data, 5 bits of CRC
=> total length of 19 bits

Slave 3: 24 bits of sensor data, 16 bits of CRC
=> total length of 40 bits

SCDATA	Addr. 0x00 . . . 0x3F							
Addr.	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0x00	Sensor data 1(7:0)							
0x01	Sensor data 1(15:8)							
0x02	0	0	0	Sensor data 1(20:16)				
0x03 . . 0x06	not changed							
0x07	0	0	CRC 1(5:0)					
0x08	Sensor data 2(7:0)							
0x09	0	0	Sensor data 2(13:8)					
0x0A . . 0x0E	not changed							
0x0F	0	0	0	CRC 2(4:0)				
0x10	Sensor data 3(7:0)							
0x11	Sensor data 3(15:8)							
0x12	Sensor data 3(23:16)							
0x13 . . 0x15	not changed							
0x16	CRC 3(7:0)							
0x17	CRC 3(15:8)							
0x18 . . 0x3F	not changed							

Table 28: Example address mapping of sensor data

In order to import new sensor data during controller accesses, iP-MB100 has two memory banks for sensor data. While sensor data is being read and written into the first RAM bank during processing of the BiSS Frame, the second RAM bank section containing sensor data of the previous frame can be read by the controller. The relevant sensor data memory banks are swapped at the end of the BiSS Frame. This can be prevented by the controller entering the command register bit HOLDBANK (see Table 81). Simultaneously the validity register (SVALIDx, Table 70) is also swapped.

REGISTER DATA

The iP-MB100 has an individual storage area used for the data of the automatic register communication. In case of a write access to the slaves the storage area must be written with the register data before starting the communication. The same memory area is also used for the read data coming from the slave during read access. The memory size of the storage area is 64 bytes.

RDATA	Addr. 0x80 . . . 0xBF; bit 7:0	R/W
Addr.	Content	
0x80	RDATA1(7:0)	
0x81 . . 0xBF	RDATA2(7:0) . . . RDATA64(7:0)	

Table 29: RDATA address range for register communication

The slave addressing IDS which is used in commands must be stored at address 0x80.

IDS	Addr. 0x80; bit 7:0	R/W
Addr.	Content	
0x80	IDS(7:0)	

Table 30: IDS for commands

CONTROL COMMUNICATION

The control frame enables protected and confirmed reading and writing of the registers of a slave. It is also used for protected and confirmed sending of commands to specified slaves or to all slaves. The control frame is compounded of 1 bit per BiSS C frame in each direction. The control bit sent from the master to the slaves is called CDM and control bit sent from the slaves to the master is called CDS. For more information about the control communication refer to the BiSS Interface PROTOCOL DESCRIPTION.

The iP-MB100 does support autonomous register communication with BiSS B and BiSS C. REGVERS is used to select the protocol type. For register write access with extended SSI the BiSS C protocol type needs to be selected.

REGVERS	Addr. 0xE5; bit 6	R/W
Code	Function	
0	Register communication BiSS A/B	
1	Register communication BiSS C	

Table 31: Type of protocol for register access

If more than one physical BiSS channel is used, the one-hot coded parameter CHSEL selects the channel that is used for control communication. For commands it is possible to run the control communication on several channels simultaneously.

CHSEL(1)	Addr. 0xE4; bit 0	R/W
Code	Function	
0	Channel 1 not used *	
1	Channel 1 used for control communication	
CHSEL(2)	Addr. 0xE4; bit 1	R/W
0	Channel 2 not used	
1	Channel 2 used for control communication †	
CHSEL(3)	Addr. 0xE4; bit 2	R/W
0	Channel 3 not used	
1	Channel 3 used for control communication †	
CHSEL(4)	Addr. 0xE4; bit 3	R/W
0	Channel 4 not used	
1	Channel 4 used for control communication †	
CHSEL(5)	Addr. 0xE4; bit 4	R/W
0	Channel 5 not used	
1	Channel 5 used for control communication †	
CHSEL(6)	Addr. 0xE4; bit 5	R/W
0	Channel 6 not used	
1	Channel 6 used for control communication †	
CHSEL(7)	Addr. 0xE4; bit 6	R/W
0	Channel 7 not used	
1	Channel 7 used for control communication †	
CHSEL(8)	Addr. 0xE4; bit 7	R/W
0	Channel 8 not used	
1	Channel 9 used for control communication †	

Table 32: Channel mapping for control communication

The selection between commands and register access is done with the parameter CTS.

CTS	Addr. 0xE5; bit 7	R/W
Code	Function	
0	Command/instruction communication	
1	Register access	
Note	The parameter CTS was called MSEL in former MB100 or iC-MB3 data sheets.	

Table 33: Type of control communication

The parameter HOLDCDM determines the behaviour of the signal MA at the end of the frame. If the signal SLx is not digital low at the end of the frame, because the BiSS C frame has not been clocked out completely, the signal MA must be programmed with HOLDCDM to be constant until the next frame begins.

* Channel 1 is selected if CHSEL(8:1)=0.
† Channel 2 ...8 are available depending on the iP-MB100 GENERICS.

HOLDCDM	Addr. 0xE5; bit 0	R/W
Code	Function	
0	MA changes to digital after detecting the slave's timeout	
1	MA remains constant until the next frame begins	

Table 34: Behaviour of the signal MA at the end of frame

The Commands

The following registers are used to configure the BiSS C control communication for command transfer. The command/instruction communication is a subset of the control communication. The command can address one or several slaves by setting the appropriate bits in the IDS register. To address all slaves via broadcast command IDS has to be reset to zero.

IDS(0:7)	Addr. 0x80; bit 7:0	R/W
Code	Function	
0x00	All slaves addressed (broadcast)	
IDS7	Addr. 0x80; bit 0	R/W
0	Slave with ID 7 is not addressed	
1	Slave with ID 7 is addressed	
IDS6	Addr. 0x80; bit 1	R/W
0	Slave with ID 6 is not addressed	
1	Slave with ID 6 is addressed	
IDS4	Addr. 0x80; bit 2	R/W
0	Slave with ID 5 is not addressed	
1	Slave with ID 5 is addressed	
IDS4	Addr. 0x80; bit 3	R/W
0	Slave with ID 4 is not addressed	
1	Slave with ID 4 is addressed	
IDS3	Addr. 0x80; bit 4	R/W
0	Slave with ID 3 is not addressed	
1	Slave with ID 3 is addressed	
IDS2	Addr. 0x80; bit 5	R/W
0	Slave with ID 2 is not addressed	
1	Slave with ID 2 is addressed	
IDS1	Addr. 0x80; bit 6	R/W
0	Slave with ID 1 is not addressed	
1	Slave with ID 1 is addressed	
IDS0	Addr. 0x80; bit 7	R/W
0	Slave with ID 0 is not addressed	
1	Slave with ID 0 is addressed	

Table 35: IDS command/instruction addressing

The command code is determined with CMD. Some commands are predefined in the BiSS Interface PROTOCOL DESCRIPTION, others can be used for application specific functionality.

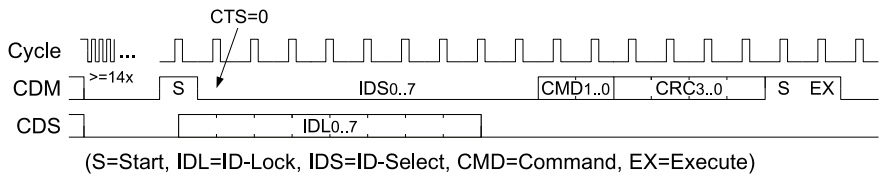
CMD(1:0)	Addr. 0xE5; bit 5:4	R/W
Code	Command / instruction	
0x00	"00"	
0x01	"01"	
0x02	"10"	
0x03	"11"	

Table 36: Command of addressed slave

In contrast to the broadcast commands, the reception of a command addressed to specific slaves is confirmed. Each addressed slave confirms the command during IDA. The parameter IDA_TEST defines, if the master triggers the command execution immediately or after successfully checking the IDA bits (see Fig. 16 and 17).

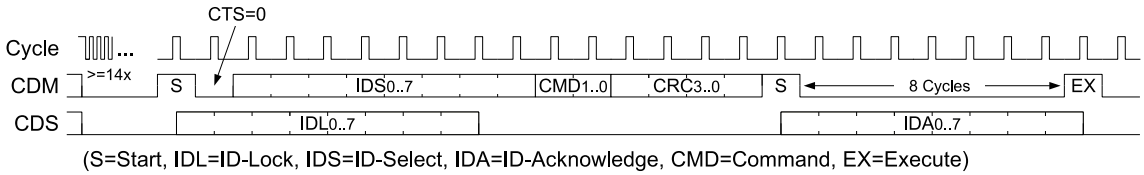
IDA_TEST	Addr. 0xE5; bit 3	R/W
Code	Function	
0	Immediate execution	
1	The slaves feedback (IDA) is tested before execution	

Table 37: Command/instruction execution control



(S=Start, IDL=ID-Lock, IDS=ID-Select, CMD=Command, EX=Execute)

Figure 16: Control frame (broadcast command)



(S=Start, IDL=ID-Lock, IDS=ID-Select, IDA=ID-Acknowledge, CMD=Command, EX=Execute)

Figure 17: Control frame (addressed command)

The register access

The register start address REGADR, register access direction RNW, number of bytes REGNUM and slave address SLAVEID are used to configure the BiSS C control communication for register access. A byte count of 0 entered for REGNUM signals the transmission of a single register value.

REGADR	Addr. 0xE2; bit 6:0	R/W
Code	Function	
0x00 . . . 0x7F	Start address for register access	

Table 38: Register access start address

WNR	Addr. 0xE2; bit 7	R/W
Code	Function	
0	Read register data	
1	Write register data	

Table 39: Register access direction

REGNUM	Addr. 0xE3; bit 5:0	R/W
Code	Register count	
0x00	1	
0x01 . . . 0x3E	Code + 1	
0x3F	64	

Table 40: Number of consecutive registers to access

SLAVEID	Addr. 0xE5; bit 5:3	R/W
Code	Slave ID	
0 . . . 7	Code	

Table 41: Slave ID of accessed slave

The Figure 18 exemplarily shows a control frame including a register read access.

Note: All configuration parameter must be stable during control communication frame.

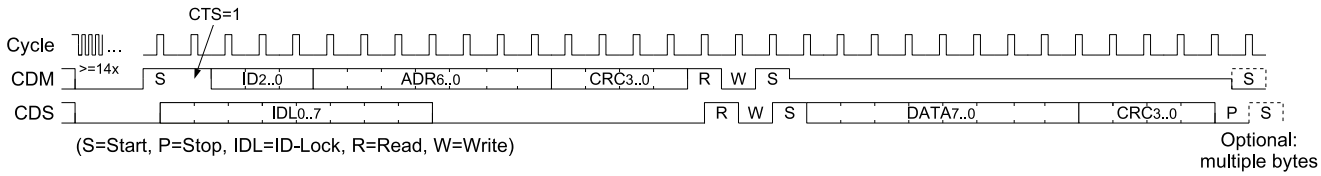


Figure 18: Control frame (register read)

SLAVE CONFIGURATION

The process data consists of a maximum of eight logical data channels. The slave configuration determines the setup of each data channel. The iP-MB100 provides a programmable data length, CRC polynomial and CRC start value for each logical data slave.

ENSCD1	Addr. 0xC0; bit 6	R/W
ENSCD2	Addr. 0xC4; bit 6	R/W
ENSCD3	Addr. 0xC8; bit 6	R/W
ENSCD4	Addr. 0xCC; bit 6	R/W
ENSCD5	Addr. 0xD0; bit 6	R/W
ENSCD6	Addr. 0xD4; bit 6	R/W
ENSCD7	Addr. 0xD8; bit 6	R/W
ENSCD8	Addr. 0xDC; bit 6	R/W
Code	Function	
0	Single cycle data not available	
1	Single cycle data available	

Table 42: Enable single cycle data for slave x

SCDLEN1	Addr. 0xC0; bit 5:0	R/W
SCDLEN2	Addr. 0xC4; bit 5:0	R/W
SCDLEN3	Addr. 0xC8; bit 5:0	R/W
SCDLEN4	Addr. 0xCC; bit 5:0	R/W
SCDLEN5	Addr. 0xD0; bit 5:0	R/W
SCDLEN6	Addr. 0xD4; bit 5:0	R/W
SCDLEN7	Addr. 0xD8; bit 5:0	R/W
SCDLEN8	Addr. 0xDC; bit 5:0	R/W
Code	Single cycle data length	
0	1	
. . .	Code + 1	
63	64	

Table 43: Single cycle data length of slave x

SELCRCS1	Addr. 0xC1; bit 7	R/W
SELCRCS2	Addr. 0xC5; bit 7	R/W
SELCRCS3	Addr. 0xC9; bit 7	R/W
SELCRCS4	Addr. 0xCD; bit 7	R/W
SELCRCS5	Addr. 0xD1; bit 7	R/W
SELCRCS6	Addr. 0xD5; bit 7	R/W
SELCRCS7	Addr. 0xD9; bit 7	R/W
SELCRCS8	Addr. 0xDD; bit 7	R/W
Code	Function	
0	CRC bit length in SCRCLENx apply predefined CRC polynomials	
1	CRC polynomial(7:1) in SCRCPOLYx	

Table 44: CRC polynomial selection

BiSS uses CRC polynomials depending on the data length predefined in the EDS and Profile Definitions. The parameter SELCRCS chooses between predefined polynomials selected according to the CRC length and non predefined polynomials for CRCs with a length of up to seven bits. If CRCLEN = 0, the CRC verification is deactivated.

SCRCLEN1	Addr. 0xC1; bit 6:0	R/W
SCRCLEN2	Addr. 0xC5; bit 6:0	R/W
SCRCLEN3	Addr. 0xC9; bit 6:0	R/W
SCRCLEN4	Addr. 0xCD; bit 6:0	R/W
SCRCLEN5	Addr. 0xD1; bit 6:0	R/W
SCRCLEN6	Addr. 0xD5; bit 6:0	R/W
SCRCLEN7	Addr. 0xD9; bit 6:0	R/W
SCRCLEN8	Addr. 0xDD; bit 6:0	R/W
Code	CRC polynomial for single cycle data	
0	CRC verification deactivated	
3	0b1011 = 0xB	
4	0b1.0011 = 0x13	
5	0b10.0101 = 0x25	
6	0b100.0011 = 0x43	
7	0b1000.1001 = 0x89	
8	0b1.0010.1111 = 0x12F	
16	0b1.1001.0000.1101.1001 = 0x190D9	
. . .	Other codes are not permitted.	

Table 45: CRC length with predefined CRC polynomial

Since the least significant bit of the CRC polynomial is always 1, there is no need to store this information in the CRC polynomial register. Therefore a maximum polynomial length of 8 bit is possible. If the maximum 8 bit polynomial length is not required, the used polynomial (without the LSB bit) is stored right aligned and leading bits are filled with 0.
Example: The CRC polynomial 0x43 = 100 0011b is stored as 010 0001b (register value 0x21).

SCRCPOLY1	Addr. 0xC1; bit 6:0	R/W
SCRCPOLY2	Addr. 0xC5; bit 6:0	R/W
SCRCPOLY3	Addr. 0xC9; bit 6:0	R/W
SCRCPOLY4	Addr. 0xCD; bit 6:0	R/W
SCRCPOLY5	Addr. 0xD1; bit 6:0	R/W
SCRCPOLY6	Addr. 0xD5; bit 6:0	R/W
SCRCPOLY7	Addr. 0xD9; bit 6:0	R/W
SCRCPOLY8	Addr. 0xDD; bit 6:0	R/W
Code	CRC polynomial for single cycle data	
0x00	CRC verification deactivated	
0x01 . . 0x7F	Code & '1'	

Table 46: CRC polynomial

The CRC start values can be used to safely differentiate between SCD data words e.g. two position words

provided by two different sensors. The same CRC start value that is used in the sensor needs to be applied in the master to validate the received single cycle data. The CRC start value has a length of 16 bit and is divided into two consecutive 8-bit registers. A typical default slave CRC start value is 0x0000.

SCRCSTART1_L	Addr. 0xC2; bit 7:0	R/W
SCRCSTART1_H	Addr. 0xC3; bit 7:0	R/W
SCRCSTART2_L	Addr. 0xC6; bit 7:0	R/W
SCRCSTART2_H	Addr. 0xC7; bit 7:0	R/W
SCRCSTART3_L	Addr. 0xCA; bit 7:0	R/W
SCRCSTART3_H	Addr. 0xCB; bit 7:0	R/W
SCRCSTART4_L	Addr. 0xCE; bit 7:0	R/W
SCRCSTART4_H	Addr. 0xCF; bit 7:0	R/W
SCRCSTART5_L	Addr. 0xD2; bit 7:0	R/W
SCRCSTART5_H	Addr. 0xD3; bit 7:0	R/W
SCRCSTART6_L	Addr. 0xD6; bit 7:0	R/W
SCRCSTART6_H	Addr. 0xD7; bit 7:0	R/W
SCRCSTART7_L	Addr. 0xDA; bit 7:0	R/W
SCRCSTART7_H	Addr. 0xDB; bit 7:0	R/W
SCRCSTART8_L	Addr. 0xDE; bit 7:0	R/W
SCRCSTART8_H	Addr. 0xDF; bit 7:0	R/W

Code	CRC start value for single cycle data
0x0000 .. 0xFFFF	SCRSTARTx_HI & CRCSTARTx_LO

Table 47: CRC calculation start value

Each data channel can be enabled to contain actuator data or sensor data. The ACTnSENS parameter defines the functionality of each BiSS slave on the BiSS bus.

ACTnSENS1	Addr. 0xEF; bit 0	R/W
ACTnSENS2	Addr. 0xEF; bit 1	R/W
ACTnSENS3	Addr. 0xEF; bit 2	R/W
ACTnSENS4	Addr. 0xEF; bit 3	R/W
ACTnSENS5	Addr. 0xEF; bit 4	R/W
ACTnSENS6	Addr. 0xEF; bit 5	R/W
ACTnSENS7	Addr. 0xEF; bit 6	R/W
ACTnSENS8	Addr. 0xEF; bit 7	R/W

Code	Function
0	Slave is configured as sensor
1	Slave is configured as actuator

Table 48: Slave functionality control

Typically, a BiSS slave expects a leading stop bit before the actuator data which must be enabled with LSTOP (refer to BiSS Interface PROTOCOL DESCRIPTION).

LSTOP1	Addr. 0xC0; bit 7	R/W
LSTOP2	Addr. 0xC4; bit 7	R/W
LSTOP3	Addr. 0xC8; bit 7	R/W
LSTOP4	Addr. 0xCC; bit 7	R/W
LSTOP5	Addr. 0xD0; bit 7	R/W
LSTOP6	Addr. 0xD4; bit 7	R/W
LSTOP7	Addr. 0xD8; bit 7	R/W
LSTOP8	Addr. 0xDC; bit 7	R/W

Code	Function
0	No leading STOP bit on single cycle actuator data
1	Leading STOP bit on single cycle actuator data

Table 49: Actuator stop bit control

The parameter GRAY activates a GRAY to binary conversion for SSI encoder.

GRAYS1	Addr. 0xC0; bit 7	R/W
GRAYS2	Addr. 0xC4; bit 7	R/W
GRAYS3	Addr. 0xC8; bit 7	R/W
GRAYS4	Addr. 0xCC; bit 7	R/W
GRAYS5	Addr. 0xD0; bit 7	R/W
GRAYS6	Addr. 0xD4; bit 7	R/W
GRAYS7	Addr. 0xD8; bit 7	R/W
GRAYS8	Addr. 0xDC; bit 7	R/W

Code	Function
0	SSI single cycle data binary coded
1	SSI single cycle data GRAY coded

Table 50: SSI format is GRAY code

MASTER CONFIGURATION

System clock

The system clock is provided by an external clock signal of 20 MHz connected to the signal CLK.

CLKENI	Addr. 0xF5; bit 0	R/W
Code	Function	
0	External oscillator via signal CLK	
1	Internal oscillator	

Table 51: System clock source

Master Clock MA

The master clock at MA for both BiSS and SSI is generated from the system clock depending on the frequency division register FREQ. With a system clock frequency of 20 MHz, the clock frequency at MA ranges from 10 MHz down to 62.5 kHz.

FREQS	Addr. 0xE6; bit 4:0	R/W
Code	Single cycle data clock frequency (FSCD)	
0x00	f_{CLK} / 2	
0x01 .. 0x0F	f_{CLK} / 2 / (Code + 1)	
0x10	"not permitted"	
0x11	f_{CLK} / 40	
0x12 .. 0x1F	f_{CLK} / 20 / (Code - 15)	

Table 52: Single cycle data clock frequency at MA

Both BiSS and SSI slave devices recognize an idle bus at the end of a transmission frame via timeout detection (see $t_{BiSS\text{-}Timeout}$ in BiSS Interface PROTOCOL DESCRIPTION). Thus, the choice of possible clock frequencies is limited as the duration of each logic level may not exceed the shortest timeout of all the connected slaves.

BiSS B devices switch to register mode when recognizing that the bus is idle after a falling-edge on the clock input MA. The master is notified about entering register mode via data line SL, see item 306 Ttos.

The clock frequency in BiSS B register mode is selected depending on the single cycle data clock frequency via parameter FREQR and is in the range of 244 Hz to 5 MHz. The selection is also limited since a different timeout detection does now recognize the idle bus at the end of the cycle (timeoutREG, see BiSS B protocol).

FREQR	Addr. 0xE6; bit 7:5	R/W
Code	Register data clock frequency	
0	FSCD / 2	
1 .. 6	FSCD / 2 / (Code+1)	
7	FSCD / 256	

Table 53: BiSS B register data frequency

BiSS B devices typically require a minimum clock frequency (such as 250 kHz) because the MA clock possibly has to be evaluated as a PWM signal for register communication. BiSS C devices generally permit a lower clock frequency. BiSS C devices do not use a MA clock duty cycle (PWM signal) and can be operated down to 80 kHz. SSI devices generally permit a lower clock frequency and with extended SSI, the register access is similar to BiSS C and can be operated down to 80 kHz.

BiSS frame rate

The FREQAGS controls the automatic data transmission (AutoGetSens) enabled by the instruction bit AGS (see Table 77). With FREQAGS the frame rate can be set to a dedicated ratio of the system clock frequency f_{CLK}. With a system clock frequency of 20 MHz frame repetition rates from 1 µs to 4 ms are possible. FREQAGS must be set in a way that the time in between two frames is greater than a complete cycle. One cycle consists of the transmission of a request, an acknowledge signal (including line delay), a start bit (including processing time), a control data bit (BiSS C), the sensor data and CRC bits of each slave and the longest BiSS timeout of all connected slaves.

FREQAGS	Addr. 0xE8; bit 7:0	R/W
Code	Register data clock frequency	
0x00	f_{CLK} / 20	
0x00 .. 0x7B	f_{CLK} / 20 / (Code +1)	
0x7C	AGSMIN	
0x7D .. 0x7F	AGSINFINITE	
0x80 .. 0xFF	f_{CLK} / 625 / (Code - 127)	

Table 54: AutoGetSens frequency

AGSMIN

With AGSMIN the master automatically restarts the next frame after the previous frame is finished. If AGSMIN is chosen the iP-MB100 automatically generates the highest frame rate possible. AGSMIN requires complete BiSS frames to ensure a low level at SL at the

end of each frame. The rate depends on the configured sensor data clock frequency, the slave configurations, the longest processing time of the slaves and the total system line delay.

AGSINFINITE
With AGSINFINITE the master does not automatically restart the next cycle. AGSINFINITE requires a trigger event to start the next frame. Possible trigger events are a digital impulse at the GETSENS signal accessing the instruction register (see Table 78).

Data handling
The received and inverted CRC bits are stored at the most significant SCDATA bits (refer to sensor and actuator data, page 18). This can be disabled with the parameter NOCRC = 1.

NOCRC	Addr. 0xE7; bit 1	R/W
Code	Function	
0	CRC of SCD is stored in process data memory	
1	CRC of SCD not stored in process data memory	

Table 55: Storage of received CRC in process data memory

Two RAM banks are available for buffering the SCDATA (refer to sensor and actuator data, page 18). A downgrade is possible with the parameter SINGLEBANK = 1.

SINGLEBANK	Addr. 0xE7; bit 1	R/W
Code	Function	
0	Two RAM banks are used for SCD	
1	One RAM bank is used for SCD	

Table 56: Usage of single RAM bank for SCD

The data output signal MOx is used to set the processing time per parameter for all slaves and to send the actuator data. The usage of all the output MOx must be enabled with EN_MO = 1.

EN_MO	Addr. 0xE5; bit 1	R/W
Code	Function	
0	Set and keep MOx to low state	
1	Data output at MOx enabled	

Table 57: Enable data output at MO

The processing time per parameter is set by the delay of the start bit at MOx. The delay is configured with MO_BUSY in periods of MA.

MO_BUSY	Addr. 0xE9; bit 7:0	R/W
Code	Start bit delay at MOx	
0x00	No start bit delay	
0x01 .. 0x0F	Code * 1 / FSCD	

Table 58: Processing time per parameter

BiSS Master Device Identification
The BiSS master device is identifiable with the two registers VERSION and REVISION. A host software can use VERSION and REVISION to identify the present device and verify the compatibility of the software and the device.

REVISION	Addr. 0xEA; bit 7:0	R
Code	Revision	
0x10	Z(first revision)	
0x11	Z1	
0x20	Y	
0x30	X (latest revision)	
...		
0xFF		

Table 59: iP-MB100 redesign ID

The iP-MB100 BiSS iP´s do also provide a version and revision for identification.

VERSION	Addr. 0xEB; bit 7:0	R
0x41	BiSS iP MB101	
0x42	BiSS iP MB102	
0x43	BiSS iP MB103	
0x44	BiSS iP MB104	
0x45	BiSS iP MB105	
0x46	BiSS iP MB106	
0x47	BiSS iP MB107	
0x48	BiSS iP MB108	
0x49	BiSS iP MB109	
0x4A	BiSS iP MB110	
0x4B	BiSS iP MB111	
0x4C	BiSS iP MB112	
0x4D	BiSS iP MB113	
0x4E	BiSS iP MB114	
0x4F	BiSS iP MB115	
0x50	BiSS iP MB116	
0x51	BiSS iP MB117	
0x52	BiSS iP MB118	
0x53	BiSS iP MB119	
0x54	BiSS iP MB120	
0x55	BiSS iP MB121	
0x56	BiSS iP MB122	
0x57	BiSS iP MB123	
0x58	BiSS iP MB124	
...	...	
0x82	BiSS iP MB1xx	

Table 60: MB100 version

CONFIGURATION CHANNELS

The iP-MB100 provides options defined in the GENERICS including the quantity BiSS channels with each BiSS signal MAx, MOx and SLx. Therefor the parameters SLAVELOCx are used to select the locations for the slaves.

SLAVELOC1	Addr. 0xEC;	bit 0	R/W
SLAVELOC2	Addr. 0xEC;	bit 1	R/W
SLAVELOC3	Addr. 0xEC;	bit 2	R/W
SLAVELOC4	Addr. 0xEC;	bit 3	R/W
SLAVELOC5	Addr. 0xEC;	bit 4	R/W
SLAVELOC6	Addr. 0xEC;	bit 5	R/W
SLAVELOC7	Addr. 0xEC;	bit 6	R/W
SLAVELOC8	Addr. 0xEC;	bit 7	R/W

Code	Function
0	Slaves are assigned to this channel
1	Following slaves are assigned to the next channel

Table 61: Enable single cycle data for slave x

The protocol for each channel is separately defined with CHSEL.

CFGCH1	Addr. 0xED;	bit 1:0	R/W
CFGCH2*	Addr. 0xED;	bit 3:2	R/W
CFGCH3*	Addr. 0xED;	bit 5:4	R/W
CFGCH4*	Addr. 0xED;	bit 7:6	R/W
CFGCH5*	Addr. 0xEE;	bit 1:0	R/W
CFGCH6*	Addr. 0xEE;	bit 3:2	R/W
CFGCH7*	Addr. 0xEE;	bit 5:4	R/W
CFGCH8*	Addr. 0xEE;	bit 7:6	R/W

Code	Function
0x00	BiSS B
0x01	BiSS C
0x02	SSI
0x03	Channel is not used (no device connected)
Notes	In previous MB100 or iC-MB3 data sheets this configuration was applied by the former parameter SELSSI and BISSMOD.

Table 62: Channel configuration

* Channel 2 . . . 8 are available depending on the iP-MB100 GENERICS.

STATUS INFORMATION

The status registers indicate all states of the master. The status information is combined in a set of register. The status contains the state of the device and communication.

The EOT flag is connected to the signal EOT and signals a running frame see chapter FUNCTIONAL DESCRIPTION on page 14.

EOT	Addr. 0xF0; bit 0	R
Code	Function	
0	Data transmission active	
1	Data transmission not active	

Table 63: End of transmission

An error in the single cycle data detected by checksum verification (CRC) is shown with nSCDERR. If a sensor data error is signaled, the faulty sensor can be verified by reading SVALIDx (see Table 70). The nSCDERR flag is set after power on and after executing INIT (see Table 79).

nSCDERR	Addr. 0xF0; bit 4	R
Code	Function	
0	Error in last single cycle data transmission	
1	No error in last single cycle data transmission	

Table 64: SCD transmission error

The control communication including the register access uses three dedicated flags. The REGEND signals a finished control communication and nREGERR signals the status. A missing start bit is shown with NDELAYERR. The REGEND flag is reset after power on and by starting a new control communication.

REGEND	Addr. 0xF0; bit 2	R
Code	Function	
0	Control communication running or not started since reset	
1	Control communication completed	

Table 65: End of register communication

The nREGERR and the nDELAYERR flags are set after power on and after executing INIT (see Table 79). If a register data error is detected, the number of bytes transmitted correctly before the error occurred is provided by the register message REGBYTES (see Table 72). In case of an error the transmission of data is terminated.

nREGERR	Addr. 0xF0; bit 3	R
Code	Function	
0	Error in last register data transmission	
1	No error in last register data transmission	

Table 66: Register communication error

nDELAYERR	Addr. 0xF0; bit 5	R
Code	Function	
0	Missing start bit in last register data transmission	
1	No error in last register data transmission	

Table 67: Start bit in register communication

An AGS watchdog error nAGSERR is set during the automatic transmission of sensor data enabled by the instruction bit AGS (see Table 77) if no new cycle could be initiated. If the last BiSS frame has not been finished in time, the next BiSS frame will be omitted. The following BiSS frame will be executed if possible. The nAGSERR flag is set when resetting the instruction bit AGS (typically by writing BREAK into the instruction register (see Table 80).

nAGSERR	Addr. 0xF0; bit 6	R
Code	Function	
0	At least one BiSS frame has been omitted	
1	No missing BiSS frames	

Table 68: AGS error

The nERR flag indicates the state of the signal NER. It is possible to connect other components to signal NER using an open collector low active error signal. Additionally the summary of all internal errors is also output to the NER signal.

nERR	Addr. 0xF0; bit 7	R
Code	Function	
0	External or internal error occured	
1	No error occurred	

Table 69: State of the signal NER

The CRC verification result of the received single cycle sensor data of every BiSS frame is written to the validity message register SVALID for each slave separately. If the CRC is disabled in the slave configuration the correspondent SVALID flag is set after the reading of the sensor data is complete. After reading the sensor data, it is recommended to reset the validity flags by writing to the SVALID register. This way, it is possible to recognize updated sensor data.

SVALID1	Addr. 0xF1; bit 1	R/W
SVALID2	Addr. 0xF1; bit 3	R/W
SVALID3	Addr. 0xF1; bit 5	R/W
SVALID4	Addr. 0xF1; bit 7	R/W
SVALID5	Addr. 0xF2; bit 1	R/W
SVALID6	Addr. 0xF2; bit 3	R/W
SVALID7	Addr. 0xF2; bit 5	R/W
SVALID8	Addr. 0xF2; bit 7	R/W
Code	Function	
0	SCD invalid	
1	SCD valid	

Table 70: SCDATAx validity indication

Switching the SCDATA RAM bank which is executed at the end of a BiSS frame does also switch the SVALIDx flags. It fails, if the user disables switching via HOLDBANK in the instruction register (see Table 81). The lastly received data are overwritten. This event is signaled with SWBANKFAILS.

SWBANKFAILS	Addr. 0xFB; bit 0	R
Code	Function	
0	Bank switching (SCD) successful	
1	Bank switching (SCD) not successful	

Table 71: Bank switching status

In case of an error in a sequential register access, the number of faultless transmitted register values are stored in REGBYTES.

REGBYTES	Addr. 0xF3; bit 5:0		R
Code	nREGERR	Number of valid register bytes	
0	1	All	
0x00 .. 0x3F	0	Code	

Table 72: Number of valid register bytes

For BiSS C control communication a minimum number of SCD cycles with exclusively CDM = 0 must be sent before starting a new control communication frame. For a manual control communication by host control the CDMTIMEOUT bit indicates that $\geq$ 14 SCD cycles with exclusively CDM = 0 have already been sent.

CDMTIMEOUT	Addr. 0xF3; bit 7	R
Code	Function	
0	CDMTIMEOUT not reached	
1	CDMTIMEOUT reached	

Table 73: CDM timeout reached

The value of the control data slave bit (CDS) is sampled for each channel in CDS. Additionally, the CDS of the selected channel (refer to Table 32, page 20) is provided by CDSSEL.

CDSSEL	Addr. 0xF3; bit 6	R
Code	CDS value from the selected channel	
0	0	
1	1	

Table 74: CDS bit from the selected channel

CDS1	Addr. 0xF8; bit 1	R
CDS2	Addr. 0xF8; bit 3	R
CDS3	Addr. 0xF8; bit 5	R
CDS4	Addr. 0xF8; bit 7	R
CDS5	Addr. 0xF9; bit 1	R
CDS6	Addr. 0xF9; bit 3	R
CDS7	Addr. 0xF9; bit 5	R
CDS8	Addr. 0xF9; bit 7	R
Code	CDS value from appropriate channel	
0	CDSx = 0	
1	CDSx = 1	

Table 75: CDSx bit of channels

Note: According to SCDATA two banks for SVALID, CDSSEL and CDS exists which switch simultaneous to the SCDATA RAM banks.

The actual state of the BiSS data input is available at SL.

SL1	Addr. 0xF8; bit 0	R
SL2	Addr. 0xF8; bit 2	R
SL3	Addr. 0xF8; bit 4	R
SL4	Addr. 0xF8; bit 6	R
SL5	Addr. 0xF9; bit 0	R
SL6	Addr. 0xF9; bit 2	R
SL7	Addr. 0xF9; bit 4	R
SL8	Addr. 0xF9; bit 6	R
Code	Function	
0	SLx line level low	
1	SLx line level high	

Table 76: SLx input lines state

INSTRUCTION REGISTER

The instruction register controls all functions based on the configuration. Data transmissions are triggered by program, by signal event or automatically. The automatic data transmission is enabled with AGS and the frame rate is set with the parameter FREQAGS (refer to Table 54, page 25).

AGS	Addr. 0xF4; bit 0	R/W
Code	FREQAGS	Function
0		No automatic data transmission
1	AGSMIN	Start of data transmission after TIMEOUTSENS
1	AGSINFINITE	Start of data transmission triggered by signal GETSENS
1	Rest	Start of data transmission equally spaced

Table 77: Automatic Get Sensor Data

The register INSTR can be used to trigger data transmission by program. With AGS = 0 the master starts the data transmission after finishing writing the instruction register (rising edge of NWR on parallel interface, last rising edge of SCLK on SPI interface) and resets INSTR after executing the frame automatically. An nAGSERR error will be generated if the SLx line is low at the start of the frame. The error can be suppressed by setting FREQAGS to AGSMIN due to waiting for a high state at SLx. A BiSS-B register access or a BiSS C control communication (refer to REGVERS, page 20) can be triggered via INSTR as well. Also, it is possible to enable reduced BiSS frames for control communication with broadcast addressing. Within a running control communication the CDM bit is generated automatically otherwise the generation of CDM depends on INSTR according to Table 78.

INSTR	Addr. 0xF4; bit 3:1	R/W
Code	AGS	Function
010	0	Start one frame with CDM = 0. INSTR automatically resets.
010	1	Upcoming frames with CDM = 0.
001	0	Start one frame with CDM = 1. INSTR automatically resets.
001	1	Upcoming frames with CDM = 1.
011	0	Start one frame with CDM = not(CDSSEL). INSTR autom. resets.
011	1	Upcoming frames with CDM = not(CDSSEL).
100	0	Start one frame and start control communication. INST autom. resets.
100	1	Upcoming frames with control communication
111	0	Start one reduced frame and control communication. INST autom. resets.
111	1	Upcoming reduced frames with control communication

Table 78: SCD Control Instruction

Note: With automatic data transmission (AGS = 1) at the end of a control communication INSTR will be reset

Typically, the BiSS data line SLx is set after timeout to digital high by the last slave. After power on or after a failure the last slave may not be defined and the SLx line is digital low. An initialisation sequence started by the instruction bit INIT can be used to initialize the slave chain. Additionally an INIT sets the status bits nSCDERR, nREGERR, nDELAYERR and nAGSERR to 1 an resets REGEND to 0.

INIT	Addr. 0xF4; bit 4	R/W
Code	REGVERS	Function
0	-	No operation
1	0	BiSS-B initialize (see)
1	1	BiSS-C initialize (see)

Table 79: Start initialisation sequence

With an initialisation sequence the iP-MB100 stores the measured channel 1 line delay in the register SCDATA1(7:0) and the channel 2 line delay in the register SCDATA5(7:0).
The unit of these values is 1/4 of the configured MA clock frequency.

$$t_{\text{Line Delay Channel 1}} = \frac{SCDATA1(7:0)}{4 * f_{MA}}$$

$$t_{\text{Line Delay Channel 2}} = \frac{SCDATA5(7:0)}{4 * f_{MA}}$$

For the init sequence the maximum line delay is 255. If exceeding this limit the init sequence is aborted and nAGSERR is reset (see 68). The init sequence does not test the SLx state. A constant SLx = 0 state is ignored and does not set any error state.

All current actions can be aborted using the BREAK command so that the iP-MB100 enters a defined state if one of the sensors does not response correctly.

BREAK	Addr. 0xF4; bit 7	R/W
Code	Function	
0	No operation	
1	Abort data transmission	
Note	An INIT should be executed after BREAK to ensure resetting status bits	

Table 80: Start BREAK Sequence

During reading of more than one sensor data register by the controller it is possible that the RAM banks in the master could be swapped once a sensor data transmission is complete. To avoid unexpected bank swaps, the controller can set HOLDBANK before the read sequence and releases it afterwards.

HOLDBANK	Addr. 0xF4; bit 6	R/W
Code	Function	
0	No bank switching permitted	
1	Bank switching permitted	

Table 81: RAM Bank Control

Every set or reset of the bit SWBANK forces the sensor data banks to be swapped.

SWBANK	Addr. 0xF4; bit 5	R/W
Code	Function	
0	No operation	
1	RAM banks are switched	

Table 82: RAM Bank Switching

The following parameters are used to control the the MA clock line which is selected with the parameter CHSEL (see Table 32).

MAFS	Addr. 0xF5; bit 4	R/W
Code	Function	
0	Selected MA line not controlled	
1	Selected MA line forced with MAVS	

Table 83: Control of the selected MA line

MAVS	Addr. 0xF5; bit 5	R/W
Code	Force value for selected MA	
0	Low	
1	High	

Table 84: Selected MA line force level

The following parameters are used to control the MA clock lines for the remaining channels which are not selected with CHSEL.

MAFO	Addr. 0xF5; bit 6	R/W
Code	Function	
0	Not selected MA lines not controlled	
1	Not selected MA lines forced with MAVO	

Table 85: Control of the not selected MA line

MAVO	Addr. 0xF5; bit 7	R/W
Code	Force value for not selected MA	
0	Low	
1	High	

Table 86: Not selected MA lines force level

GETSENS CONTROL

The signal GETSENS can be used to start a SCD cycle. The default signal state of GETSENS is digital low due to the pull down resistor at this signal. The GETSENS signal operates statically: A digital high at GETSENS will start a SCD cycle. The minimum signal duration at the GETSENS signal is one system clock period. A digital high that is longer than the SCD cycle will restart the next SCD cycle.

For high precision capturing it is recommended to use a system clock based GETSENS digital high pulse.

DESIGN REVIEW

Function Notes

iP-MB100_Z1		
No.	Function, Parameter/Code	Description and application notes
1	**MO_BUSY**	Control communication max fail, if MO_BUSY /= 0 and output MO is **not** connected to slave input SLI.
2	**Actuator data**	Transmission of actuator data on both channels with different SCD length fail, if CDM = 0.
3	**BiSS C initialize sequence**	Line delay measurement fails, if line delay of one channel is bigger than line delay plus timeout of the other channel.

Table 87: Notes on iP functions regarding iP-MB100 release Z1.

All listed chip functions are adjusted with iP-MB100 release Y.

iP-MB100_Y		
No.	Function, Parameter/Code	Description and application notes

Table 88: Notes on iP functions regarding iP-MB100 chip release Y.

iP-MB100_X		
No.	Function, Parameter/Code	Description and application notes
1	SCD CRC value and NOCRC parameter	Read CRC value is accessible in SCDATAx with NOCRC = 0

Table 89: Notes on iP functions regarding iP-MB100 release X.

REVISION HISTORY

Rel.	Rel. Date*	Chapter	Modification	Page
A1.0	2006-12-13	Initial release		all

Rel.	Rel. Date*	Chapter	Modification	Page
A1.1	2007-01-25		PDF secured	all

Rel.	Rel. Date*	Chapter	Modification	Page
A1.2	2008-07-01		Minor text updates	all

Rel.	Rel. Date*	Chapter	Modification	Page
A1.3	2008-07-10		Minor updates English release	all

Rel.	Rel. Date*	Chapter	Modification	Page
A1.4	2008-07-10		Minor updates German release	

Rel.	Rel. Date*	Chapter	Modification	Page
A1.5	2011-06-01		Description of MO functions added: EnMO, MO BUSY	

Rel.	Rel. Date*	Chapter	Modification	Page
A 1.6	2011-12-13		Minor updates German release	

Rel.	Rel. Date*	Chapter	Modification	Page
A1.7	2012-01-25		Minor updates English release	

Rel.	Rel. Date*	Chapter	Modification	Page
A1.8	2012-08-02		Minor updates English release	
			German releases discontinued	
			RVALIDx changed into CDSx	
			VALID changed into SVALID	
			CDMTIMEOUT changed into CDMTO	
			BREAK = 1 stops active communication	
			FREQR(2:0) and FREQ(4:0) in address 0xE6 updated	

Rel.	Rel. Date*	Chapter	Modification	Page
B1	2018-11-26		iP-MB100_X (iP-MB100 release X) major update	

* Release Date format: *YYYY-MM-DD*

ORDERING INFORMATION

Type	Package	Options	Order Designation
iP-MB101	VHDL	Valid ”BiSS Device Manufacturer” license or ”BiSS iC iP Provider” license required.	iP-MB101

Please send your purchase orders to our order handling team:

Fax: +49 (0) 61 35 - 92 92 - 692
E-Mail: dispo@ichaus.com

For technical support, information about prices and terms of delivery please contact:

iC-Haus GmbH
Am Kuemmerling 18
D-55294 Bodenheim
GERMANY

Tel.: +49 (0) 61 35 - 92 92 - 0
Fax: +49 (0) 61 35 - 92 92 - 192
Web: http://www.ichaus.com
E-Mail: sales@ichaus.com

Appointed local distributors: http://www.ichaus.com/sales_partners