iC-MB100
BiSS INTERFACE MASTER

PRELIMINARY CONFIDENTIAL

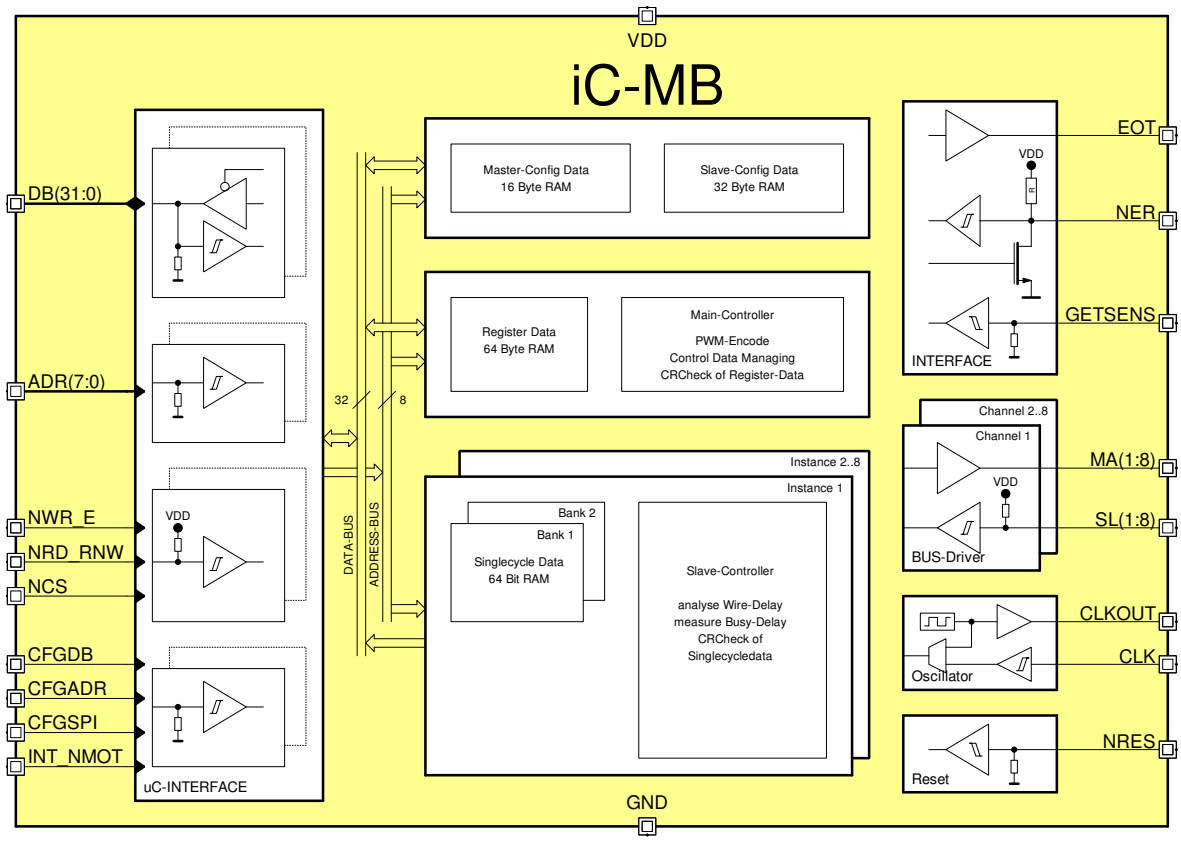
FEATURES

- Bi-directional *BiSS* communication with up to 8 sensors
- Supports SSI protocol for uni-directional data transmission
- Synchronous sensor data acquisition with cyclic transfer at data rates of up to 10 Mbit/s
- Command and slave register operations during cyclic data transfers
- Automatic compensation of line delays, measurement and conversion times
- Data lengths of up to 64 bits for sensor data, independently scalable for each slave
- Data verification by CRC polynomials of up to 8 bits, adjustable per slave and data area
- Separate memory banks enable free controller access during *BiSS* sensor data transfers
- Parallel controller interface with an 8, 16 or 32 bit data/address bus services Intel and Motorola devices with separate 8 bit adress bus or combined data and adress bus
- Serial controller communication by SPI™-compatible mode

APPLICATIONS

- Bi-directional communication in multi-sensor systems
- Linear and rotary encoders
- Motor-feedback systems

BLOCK DIAGRAM



DESCRIPTION

iC-MB100 representates a group of similar *BiSS* master devices based on the same structures and functionalities as are iC-MB101, iC-MB105, .. .

iC-MB100 allows microcontroller communication of the Intel 8051 family or Motorola 68HC11 with *BiSS* or SSI devices. An additional SPI interface mode also enables serial communication.

Up to eight *BiSS* devices can be attached to the actuator/sensor side of the device. These are connected to clock lines MA1 to MA8 and data return line SL1 to SL8 using RS422 transceivers. In noise-free environments the *BiSS* devices can also be connected directly.

A maximum of eight *BiSS* slaves is supported, each with their own independently scalable data sections encompassing:

1. Sensor data from 0 to 64 bits
   (for measurement data, alarms and warnings)
2. Register data with 128 bytes per slave ID
   (e.g. for device parameters)

For single cycle data iC-MB100 provides dual RAM memory banks for each slave, enabling flexible access to the microcontroller while new sensor data is being read in. A 64 byte buffer memory supports register transfers.

Sensor data acquisition is started by a microcontroller command or via pin GETSENS. Alternatively, iC-MB100 can also read in new sensor data automatically; the cycle time in this instance can be set as required.

The end of sensor data acquisition and read-in is signalled at pin EOT by a *high* level; if errors occur during transmission pin NER signals a *low*. Errors in communication can be verified by the microcontroller via a status register; a system error message can also be shown in this register, if bi-directional message pin NER is kept *low* externally.

INTERFACE

PIN FUNCTIONS

No.	Name	Function
	NWR_E	Write Input, low active (Intel) Enable Input, high active (Motorola)
	NRD_RNW	Read Input, low active (Intel) Read/Not-Write Select Input (Motorola)
	NCS	Chip Select Input, low active
	ADR0_ALE	Address Bus or Address Latch Enable Input or SPI Clock Input
	ADR*i*	Address Bus
	DB0	Data Bus (bidirectional) or SPI Serial Data Input
	DB1	Data Bus (bidirectional) or SPI Serial Data Output
	DB*i*	Data Bus (bidirectional)
	DBIN	Data Bus Input

PIN FUNCTIONS

No.	Name	Function
	DBOUT	Data Bus Output
	INT_NMOT	Mode Select (Intel = 1, Motorola = 0)
	CFGDB	Data Bus Width Select Input
	CFGADR	Address Bus Select Input
	CFGSPI	Serial/Parallel Mode Select Input (serial SPI = 1, parallel = 0)
	EOT	End-Of-Transmission Output
	GETSENS	Sensor Data Request Input
	NER	Error Message Input/Output, low active
	MA*i*	BiSS Clock Line Output
	SL*i*	BiSS Data Line Input
	NRES	Reset Input, low active
	CLK	External Clock Input
	CLKOUT	Clock Output

FILE LIST

biss.vhd	includes packege biss_synth for synthesis and biss_sim for simulation
lpm_pack.vhd	used in biss_lpm
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
mb3.vhd	Interface MA and SL
mbz402.vhd	SPI interface
mbz401.vhd	µC interface logic
mb4.vhd	µC interface
mb5.vhd	Oscillator
mb6.vhd	Power supply reset
mb7.vhd	Config RAM
mb8.vhd	Config Single cycle data
mb9.vhd	RAM for Singe cycle data
mba.vhd	RAM for RegisterData
mbb.vhd	Channel select
mbc.vhd	Interface divers
mb0.vhd	configurable topcell
mb101.vhd	application specific topcell

The following files includes description for RAMs. Only one of them should be used, depending on FPGA device and manufacturer.

biss_lpm.vhd	use for Altera ACEX1
biss_altera.vhd	use for Altera CYCLONE
biss_xilinx.vhd	use for Xilinx
biss_ProASIC3E.vhd	use for Actel ProASIC3E
biss_structure.vhd	use for the rest

Example TCL-scripts for project setup and compilation.

mb101.quartus.tcl	for use with Altera Quartus
mb101.ise.tcl	for use with Xilinx Ise

CONFIGURATION

Generics ..... page 4

BusWidth:	Data bus width
Slaves:	Slave count
Channels:	Channel count
AddressBus:	Adress bus configuration
UseLatch:	Latch configuration
UseTristate:	Bus tristate configuration
NumRegs:	Register count
UseSPI:	Use serial interface
UseARes:	RAM configuration
BankCount:	Bank count
RamWidth	RAM data bus width
UseFpgaRam	FPGA RAM selector
ClkdivSensLen:	Clock counter sensor data frequency
ClkdivAgsLen:	Clock counter automatic sensor data request
ClkdivRegLen:	Clock counter register data frequency
BusyCountLen:	Clock counter supervising process time
UseReg245:	Enable test register 245
Revision:	Revision
Version:	Device identifier

GENERICS

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 und DB16 = 0: Data bus width 16 Bit CFGDB = 1 und DB16 = 1: Data bus width 8 Bit

BusWidth, RamWidth		
8, 8		Data bus width 8 Bit
16, 16		Data bus width 16 Bit
16, 8	CFGDB = 0	Data bus width 16 Bit
16, 8	CFGDB = 1	Data bus width 8 Bit
32, 32		Data bus width 32 Bit
32, 16	CFGDB = 0	Data bus width 32 Bit
32, 16	CFGDB = 1	Data bus width 16 Bit
32, 8	CFGDB = 0	Data bus width 32 Bit
32, 8	CFGDB = 1 und DB16 = 0	Data bus width 16 Bit
32, 8	CFGDB = 1 und DB16 = 1	Data bus width 8 Bit

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

Channels	
1... 8	*BiSS* channel count

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

UseLatch	
0	Use synchronous RAMs (flip-flops)
1	Use asynchronous RAMs (latches)

UseTristate	
0	Separate read/write data bus
1	Single bidirectional data bus

NumRegs	
1... 64	Byte count for register communication

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

UseARes	
0	Use RAMs without asynchronous reset
1	Use RAMs with asynchronous reset

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

UseFpgaRam	
0	No RAM
1	Use single-port RAM
2	Use dual-port RAM

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

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) $*$ $2^1...2^7$

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

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

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

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

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

PROGRAMMING

REGISTER LAYOUT, OVERVIEW

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	Reserved							
Register Data								
0x80	RDATA1(7:0)							
0x81 .. 0xBF	RDATA2(7:0) .. RDATA64(7:0)							
Slave Configuration								
0xC0	GRAYS1	ENSCD1	SCDLEN1(5:0)					
0xC1	INVCRCS1	SCRCPOLY1(7:1)						
0xC4 .. 0xDF	CFGSLAVE2(15:0) .. CFGSLAVE8(15:0)							
Register Communication Configuration								
0xE0	Reserviert							
0xE1	Reserviert							
0xE2	WNR	REGADR(6:0)						
0xE3	Reserved	Reserved	REGNUM(5:0)					
0xE4	CHSEL(8:1)							
0xE5	CTS	REGVERS	SLAVEID(2:0)			Reserved	EnMO	HOLDCDM
Master Configuration								
0xE6	FREQR(2:0)			FREQ(4:0)				
0xE7	Reserved							
0xE8	FREQAGS(7:0)							
0xE9	MO_BUSY(7:0)							
0xEA	REVISION(7:0)							
0xEB	VERSION(7:0)							
Channel Configuration								
0xEC	SLAVELOC(8:1)							
0xED	SELSSI4	BISSMOD4	SELSSI3	BISSMOD3	SELSSI2	BISSMOD2	SELSSI1	BISSMOD1
0xEE	SELSSI8	BISSMOD8	SELSSI7	BISSMOD7	SELSSI6	BISSMOD6	SELSSI5	BISSMOD5
0xEF	ACTnSENS(8:1)							

OVERVIEW								
Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Status Information								
0xF0	nERR	nWDERR	'1'	nSCDERR	nREGERR	REGEND	'1'	EOT
0xF1	SVALID4	'0'	SVALID3	'0'	SVALID2	'0'	SVALID1	'0'
0xF2	SVALID8	'0'	SVALID7	'0'	SVALID6	'0'	SVALID5	'0'
0xF3	CDMTO	CDS	REGBYTES(5:0)					
Instruction Register								
0xF4	BREAK	HOLDBANK	SWBANK	INIT	INSTR(2:0)			AGS
0xF5	MAVO	MAFO	MAVS	MAFS	Reserved	Reserved	Reserved	Reserved
0xF6	Reserved							
0xF7	Reserved							
Status Information 2								
0xF8	CDS4	SL4	CDS3	SL3	CDS2	SL2	CDS1	SL1
0xF9	CDS8	SL8	CDS7	SL7	CDS6	SL6	CDS5	SL5
0xFA	Reserved							
0xFB	Reserved							SWBANK-FAILS
Reserved								
0xFC .. 0xFF	Reserved							

Table 1: Register layout

Reserved registers
Reserved registers need to be written with 0x00 and may result undefined values due to GENERIC settings.

SENSOR AND ACTUATOR DATA

Adr	Content
0x00	SCDATA 1(7:0)
0x01	SCDATA 1(15:8)
0x02	SCDATA 1(23:16)
0x03	SCDATA 1(31:24)
0x04	SCDATA 1(39:32)
0x05	SCDATA 1(47:40)
0x06	SCDATA 1(55:48)
0x07	SCDATA 1(63:56)
0x08 . . 0x0F	SCDATA 2(63:0)
0x10 . . 0x17	SCDATA 3(63:0)
0x18 . . 0x1F	SCDATA 4(63:0)
0x20 . . 0x27	SCDATA 5(63:0)
0x28 . . 0x2F	SCDATA 6(63:0)
0x30 . . 0x37	SCDATA 7(63:0)
0x38 . . 0x3F	SCDATA 8(63:0)

Table 2: Adress mapping of actuator/sensor data

REGISTER DATA

Adr	Content
0x80 . . 0xBF	RDATA: Register data (64 bytes, bidirectional)

Table 3: Adress mapping register data

GENERIC configuration for register data
With the IP GENERICs the size of the register data can be configured (NumRegs):

- Minimum 1 byte register data
- Maximum 64 byte register data

Not defined register data addresses may not be programmed/written into and may result not defined return values when read.

STATUS INFORMATION 1

EOT	Addr. 0xF0; bit 0	R - 1
0	Data transmission active	
1	Data transmission finished	

REGEND	Addr. 0xF0; bit 2	R - 0
0	No valid register-data available	
1	Register-data transmission completed	

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

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

nWDERR	Addr. 0xF0; bit 6	R - 1
0	Watchdog error	
1	No watchdog error	

nERR	Addr. 0xF0; bit 7	R - 1
0	Error	
1	No error	

SVALID1	Addr. 0xF1; bit 1	R - 1
SVALID2	Addr. 0xF1; bit 3	R - 1
SVALID3	Addr. 0xF1; bit 5	R - 1
SVALID4	Addr. 0xF1; bit 7	R - 1
SVALID5	Addr. 0xF2; bit 1	R - 1
SVALID6	Addr. 0xF2; bit 3	R - 1
SVALID7	Addr. 0xF2; bit 5	R - 1
SVALID8	Addr. 0xF2; bit 7	R - 1
0	Single cycle data invalid	
1	Single cycle data valid	

CDS	Addr. 0xF3; bit 6	R - 1
0	CDS of the CHSEL selected channel = 0	
1	CDS of the CHSEL selected channel = 1	

CDMTO	Addr. 0xF3; bit 6	R - 1
0	Control data timeout not served	
1	Controldata timeout served	

INSTRUCTION REGISTER

Data transmission can be started by programming the instruction register, automatically with constant cycle time or with the external pin GETSENS. The CDM bit (end of cycle low or high) is generated by the following table or generated automatically within a running register communication.

INSTR	Addr. 0xF4; bit 3:1	R/W - 000
010	CDM=0	
001	CDM=1	
011	CDM=not(REG)	
000, 101	CDM=1 Condition: CDMTIMEOUT=1	
100	Register communication Condition: CDMTIMEOUT=1 Use to start one single cycle data cycle incl. register communication	
110	Register communication Condition: CDMTIMEOUT=1	
111	Register communication (reduced protocoll) Condition: CDMTIMEOUT=1	

With AGS=0 the master starts the data transmission after finishing writing the instruction register (rising edge of NWR). A watchdog error will be generated if the SL line is low, TIMEOUTSENS has not exceeded.

The error can be inhibited by setting FREQAGS to AGSMIN.

AGS	Addr. 0xF4; bit 0	R/W - 0
0	No automatic data transmission	
1	Start of data transmission after TIMEOUTSENS Condition: FREQAGS=AGSMIN	
1	Start of data transmission triggered by pin Condition: FREQAGS=AGSINFINITE	
1	Start of data transmission after timeout	

INIT	Addr. 0xF4; bit 4	R/W - 0
0	No changes on the data channel	
1	Initialize data channel	

BREAK = 1 aborts the active data transmission. Additionally all status information will be reset.

BREAK	Addr. 0xF4; bit 9	R/W - 0
0	No change	
1	Abort data transmission nSCDERR, nREGER nWDERR = 1 REGEND = 0	

HOLDBANK	Addr. 0xF4; bit 6	R/W - 0
0	No Bank switching lock permitted	
1	Bank switching lock permitted	

SWBANK	Addr. 0xF4; bit 5	R/W - 0
0	RAM banks are not switched	
1	RAM banke are switched	

MAFS	Addr. 0xF5; bit 4	R/W
0	Controlling selected(CHSEL) MA clock line: using MA signal	
1	Controlling selected(CHSEL) MA clock line: using MAVS level	

MAVS	Addr. 0xF5; bit 5	R/W
0	Low definition of selected(CHSEL) MA clock lines	
1	High definition of selected(CHSEL) MA clock lines	

MAFO	Addr. 0xF5; bit 6	R/W
0	Controlling unselected(CHSEL) MA clock line: using MA signal	
1	Controlling unselected(CHSEL) MA clock line: using MAVS level	

MAVO	Addr. 0xF5; bit 7	R/W
0	Low definition of unselected(CHSEL) MA clock lines	
1	High definition of unselected(CHSEL) MA clock lines	

CONFIGURATION MASTER

FREQ	Addr. 0xE6; bit 4:0	R/W
0x00 .. 0x0F	f_{CLK}/ (2 * (FREQ(3:0)+1))	
0x10	not permissable	
0x11 .. 0x1F	f_{CLK}/ (20 * (FREQ(3:0)+1))	

Table 4: Sensor data frequency

FREQR	Addr. 0xE6; bit 7:5	R/W
0	FreqSens / 2	
1	FreqSens / 4	
2	FreqSens / 8	
3	FreqSens / 16	
4	FreqSens / 32	
5	FreqSens / 64	
6	FreqSens / 128	
7	FreqSens / 256	

Table 5: Register data frequency

FREQAGS	Addr. 0xE8; bit 7:0	R/W
0x00 .. 0x7B	f_{CLK}/ (20 * (FREQAGS(6:0)+1))	
0x7C	AGSMIN	
0x7D .. 0x7F	AGSINFINITE	
0x80 .. 0xFF	f_{CLK}/ (625 * (FREQAGS(6:0)+1))	

Table 6: AutoGetSens frequency

MO_BUSY	Addr. 0xE9; bit 7:0	R/W
0x00	no start bit delay	
0x01 .. 0xFF	Start bit delay (count of MA clocks)	

Table 7: Start bit delay

REVISION	Addr. 0xEA; bit 7:0	R
0x00	First revision	
0x01 .. 0xFF	Revision number	

Table 8: Redesign ID

VERSION	Addr. 0xEB; bit 7:0	R
0x40 .. 0x4F	MB100 . . . MB163	
0x83	iC-MB3 TSSOP24	

Table 9: Device ID

CONFIGURATION CHANNEL

SLAVELOC	Addr. 0xEC; bit 7:0	R/W
	0	1
Bit 0	not permissable	Slaves 1 on channel 1
Bit 1	Slave 2 on channel 1	Slaves 2 on channel 2
Bit 2	Slave 3 on the same channel as slave 2	slave 3 on the next channel
Bit 3	Slave 4 on the same channel as slave 3	slave 4 on the next channel
Bit 4	Slave 5 on the same channel as slave 4	slave 5 on the next channel
Bit 5	Slave 6 on the same channel as slave 5	slave 6 on the next channel
Bit 6	Slave 7 on the same channel as slave 6	slave 7 on the next channel
Bit 7	Slave 8 on the same channel as slave 7	slave 8 on the next channel

BISSMOD1	Addr. 0xED; bit 0	R/W
BISSMOD2	Addr. 0xED; bit 2	R/W
BISSMOD3	Addr. 0xED; bit 4	R/W
BISSMOD4	Addr. 0xED; bit 6	R/W
BISSMOD5	Addr. 0xEE; bit 0	R/W
BISSMOD6	Addr. 0xEE; bit 2	R/W
BISSMOD7	Addr. 0xEE; bit 4	R/W
BISSMOD8	Addr. 0xEE; bit 6	R/W
0	BiSS model A/B on channel 1 . . . 8	
1	BiSS model C on channel 1 . . . 8	

SELSSI1	Addr. 0xED; bit 1:0	R/W
SELSSI2	Addr. 0xED; bit 3:2	R/W
SELSSI3	Addr. 0xED; bit 5:4	R/W
SELSSI4	Addr. 0xED; bit 7:6	R/W
SELSSI5	Addr. 0xEE; bit 1:0	R/W
SELSSI6	Addr. 0xEE; bit 3:2	R/W
SELSSI7	Addr. 0xEE; bit 5:4	R/W
SELSSI8	Addr. 0xEE; bit 7:6	R/W
0	BiSS on channel 1 . . . 8	
1	SSI on channel 1 . . . 8	

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
0	Sensor data slave 1 . . . 8	
1	Actuator data slave 1 . . . 8	

CONFIGURATION SLAVE

Single cycle data configuration

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
0	Singlecycledaten not available	
1	Singlecycledaten available	

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
0x00 . . 0x3F	Single cycle data length = SCDLENx(5:0)+1	

The length of data minus 1 needs to be set, e.g. for 64 data bits you need to enter 63.

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
0	Single cycle data binary	
1	Single cycle data gray	

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
0x00	CRC for single cycle data not present	
0x01 . . 0x7F	CRC polynome for single cycle data = SCRCPOLYx(6:0) = CRC polynome(7:1) Range of CRC polynomes = 0x01, 0x03, 0x05, 0x07, 0x09. . . 0xFF (only odd polynomes possible)	

To disable CRC verification on Single Cycle Data transfer, set SCRCPOLYx to 0x00 and extend the SCD length by the expected count of CRC bits. As the least significant bit of the CRC polynomes is always 1, there is no need to store this in the polynome register. With this a maximum polynome length if 8 bit is possible. If the maximum 8 bit polynome length is not required, the used polynome (without the lsb bit) is stored right aligned and leading bits are filled with 0.

Example: the CRC polynome 0x43 = 100 0011b is stored as 0010 0001b.

INVCRCS1	Addr. 0xC1; bit 7	R/W
INVCRCS2	Addr. 0xC5; bit 7	R/W
INVCRCS3	Addr. 0xC9; bit 7	R/W
INVCRCS4	Addr. 0xCD; bit 7	R/W
INVCRCS5	Addr. 0xD1; bit 7	R/W
INVCRCS6	Addr. 0xD5; bit 7	R/W
INVCRCS7	Addr. 0xD9; bit 7	R/W
INVCRCS8	Addr. 0xDD; bit 7	R/W
0	CRC for single cycle data not inverted	
1	CRC for single cycle data inverted	

CONFIGURATION REGISTER COMMUNICATION

REGADR	Addr. 0xE2; bit 6:0	R/W
0x00 . . 0x7F	First register adress to access	

REGNUM	Addr. 0xE3; bit 5:0	R/W
0x00	Register count = 1	
0x01 . . 0x3F	Register count = REGNUM(5:0)+1	

WNR (WriteNotRead)	Addr. 0xE2; bit 7	R/W
0	Read register data	
1	Write register data	

CHSEL	Addr. 0xE4; bit 7:0	R/W
0x00	With one existing channel this channel 1 is selected.	
0x01	Channel = 1	
0x02	Channel = 2	
0x04	Channel = 3	
0x08	Channel = 4	
0x10	Channel = 5	
0x20	Channel = 6	
0x40	Channel = 7	
0x80	Channel = 8	

SLAVEID	Addr. 0xE5; bit 5:3	R/W
0b000	SlaveID = 0	
0b001	SlaveID = 1	
0b010	SlaveID = 2	
0b011	SlaveID = 3	
0b100	SlaveID = 4	
0b101	SlaveID = 5	
0b110	SlaveID = 6	
0b111	SlaveID = 7	

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

CTS	Addr. 0xE5; bit 7	R/W
0	Instruction	
1	Register communication	

HOLDCDM	Addr. 0xE5; bit 0	R/W
0	Clock line high at end of cycle	
1	Clock line constant at end of cycle	

EN_MO	Addr. 0xE5; bit 1	R/W
0	MO to low	
1	Parameterized processing time by master on MO signal active (length: MO_BUSY).	

MO_BUSY	Addr. 0xE9; bit 7:0	R/W
0x00 .. 0xFF	Count of MA clocks as the parameterized processing time by master on MO signal. **Premise:** EN_MO = 1	

STATUS INFORMATION 2

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

CDS1 Control Data Bit Slave	Addr. 0xF8; bit 1	R - 1
CDS2 Control Data Bit Slave	Addr. 0xF8; bit 3	R - 1
CDS3 Control Data Bit Slave	Addr. 0xF8; bit 5	R - 1
CDS4 Control Data Bit Slave	Addr. 0xF8; bit 7	R - 1
CDS5 Control Data Bit Slave	Addr. 0xF9; bit 1	R - 1
CDS6 Control Data Bit Slave	Addr. 0xF9; bit 3	R - 1
CDS7 Control Data Bit Slave	Addr. 0xF9; bit 5	R - 1
CDS8 Control Data Bit Slave	Addr. 0xF9; bit 7	R - 1
0	CDS = 0	
1	CDS = 1	

SWBANKFAILS Bank switching single cycle data not successful	Addr. 0xFB; bit 0	R - 1
0	Bank switching (SCD) successful	
1	Bank switching (SCD) not successful	

FUNCTION DESCRIPTION BiSS

BiSS Frame

BiSS uses in the point-to-point configuration a clock line (MA) from the master to the slave and a data line (SL) from the slave to the master. A device contain mave multiple slaves wherat the data input (SLI) of the last slave is set to low and the slaves are daisychained (SLO → SLI) and the data output (SLO) of the first slave is directed to the master. A data line from the master to the slave is not present,

At the end of the cycle the master sends the CDM inverted on the MA clock line. After detecting the slaves timeout with SLO = 1 the master changes the MA clock line state to the high. If the BiSS frame has not been clocked out finally, e.g. for a faster configuration phase and higher control data transmission rates, the HOLDCDM needs to be set to "1" to keep the clock line constant until the next cycle starts.

The difference is explained by the figures 1 und 2.

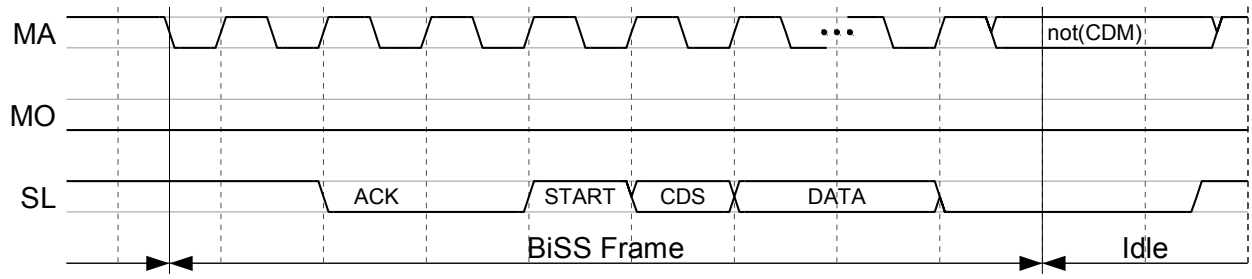


Figure 1: BiSS frame in Point-to-Point configuration (EnMO = 0, HOLDCDM = 0)

BiSS C provides the additional bus configuration with the data output line (MO) from the master to the slaves. With the EnMO = 1 the master emulates a slave without sensor data at the MO line. The parameterized processing time for sensor data that is the "start bit delay" is configured by the MO_BUSY parameter. After deactivating EnMO (only permitted to update while idle state) an INIT needs to be executed.

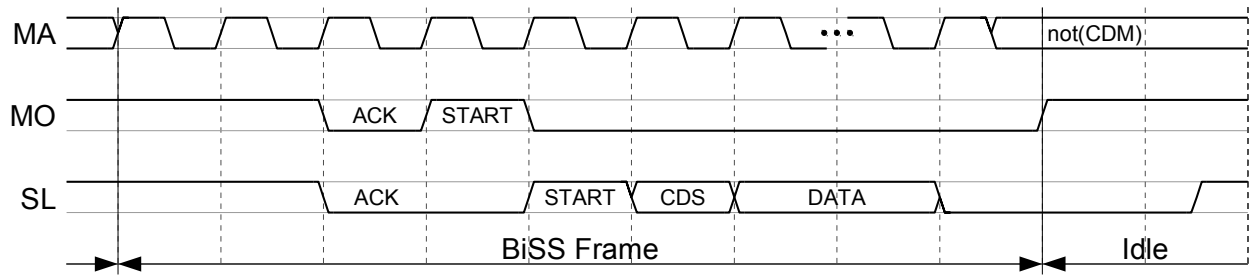


Figure 2: BiSS frame in bus configuration (EnMO = 1, HOLDCDM = 1)

2-wire / BiSS B register communication

In 2-wire / BiSS B the register communication is started by a timeing condition and a handshake at the beginning of the cycle. Alternatively a selection can be done at the cycle start with the MO line. With EnMO = 1 the slave ID ”0” stays unused (see figure 3.

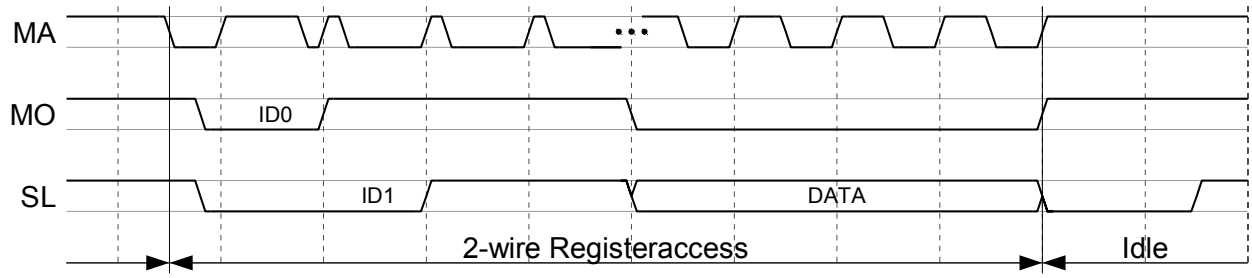


Figure 3: 2-wire / BiSS-B register access

FUNCTION DESCRIPTION SSI

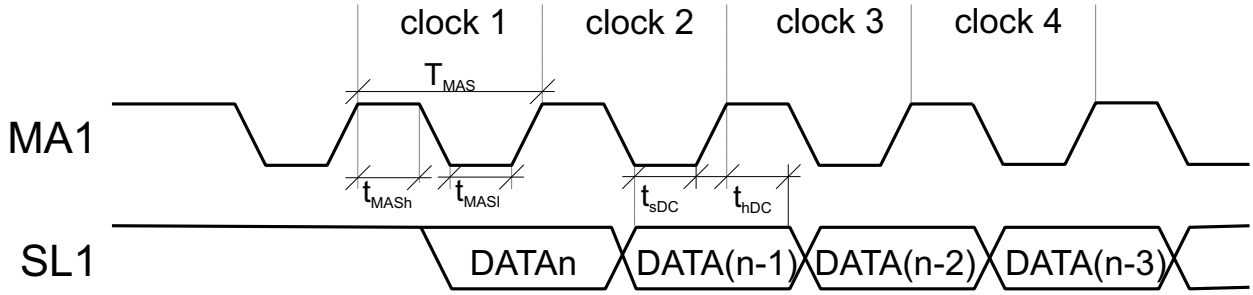


Figure 4: SSI Frame

Processing of the SL1 signals

In the SSI mode of the BiSS interface master the SL1 values are sampled with the rising edge on MA1.

Up to one MA1 period of total delay is permitted for the sensors response on SL1. This delay may be caused by in sensor processing time and transmission line delay.