

DATA SHEET



**Cologne
Chip
Designs**

HFC 2BS0

ISDN HDLC FIFO Controller

Copyright 1994, 1995 Cologne Chip Designs
All Rights Reserved

The information presented can not be considered as
assured characteristics. Data can change without notice.
Parts of the information presented may be protected by
patent or other rights.

1. Edition: October 1994
2. Edition: May 1995

Features

- Dual Read and Write HDLC-Channels for 2 ISDN B-channels
- B1 and B2 transparent mode independently selectable
- FIFO-depth: 4x 7,5 KByte (32 KByte ext. SRAM) or 4x 1,5 KByte (8 KByte ext. SRAM)
- max. 31 HDLC frames per channel and direction in FIFO
- 8 bit ISA-PC bus interface with direct drive capability (no 74xx245 needed)
- One of 6 interrupt channels on PC-ISA bus selectable
- Timer 25/50 ms with interrupt capability
- Only 2 I/O-addresses used on ISA bus
- I/O-addresses programmable
- direct connectable with ISACSTTM from SIEMENS
- 3-5V supply voltage
- rectangular QFP 80

1 General Description

The HFC is a ISDN support device for so called „passive“ ISDN PC cards. It only needs an external SRAM and a S0 interface device like the SIEMENS ISACSTTM to form a high performance ISDN PC card. Most problems with passive ISDN PC cards as small FIFOs and massive interrupt load for the host CPU are overcome by the HFC. So we call ISDN cards with the HFC „semi-active“.

The FIFOs of the HFC are realized with an external SRAM. Data bus is shared between SRAM and S0 device. An ISA-PC bus interface is also added to implement easily an ISDN PC-card.

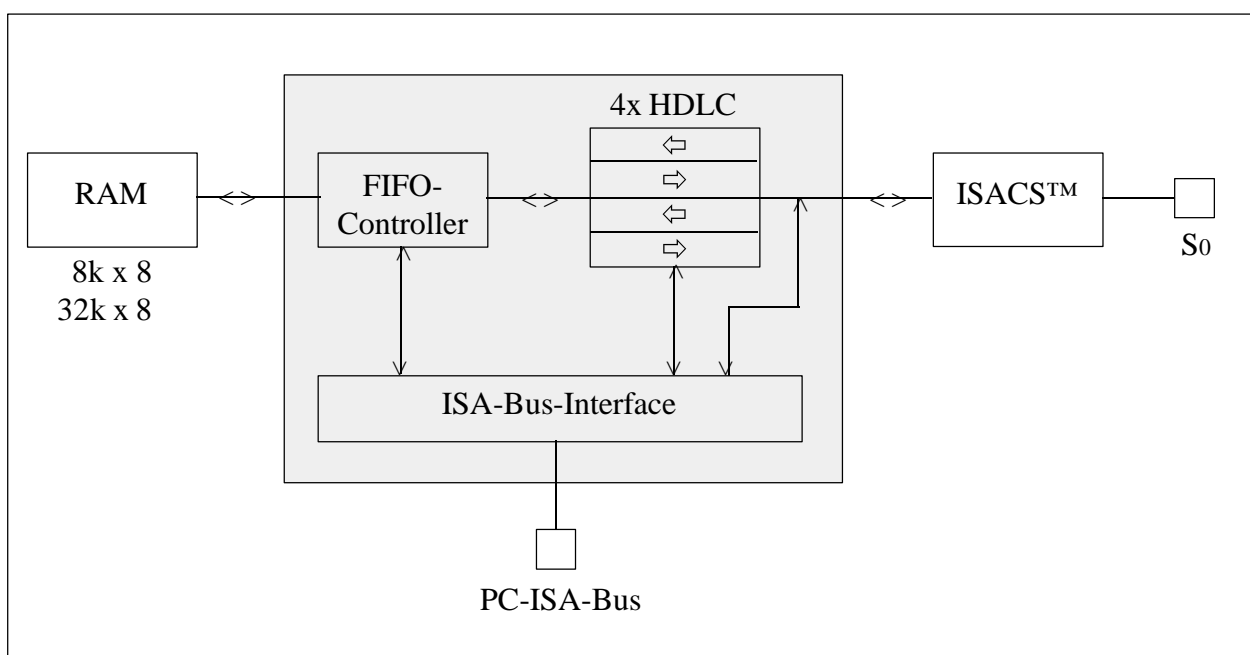


Figure 1: HFC Block Diagramm for semi-active PC-card

2 Pin Description

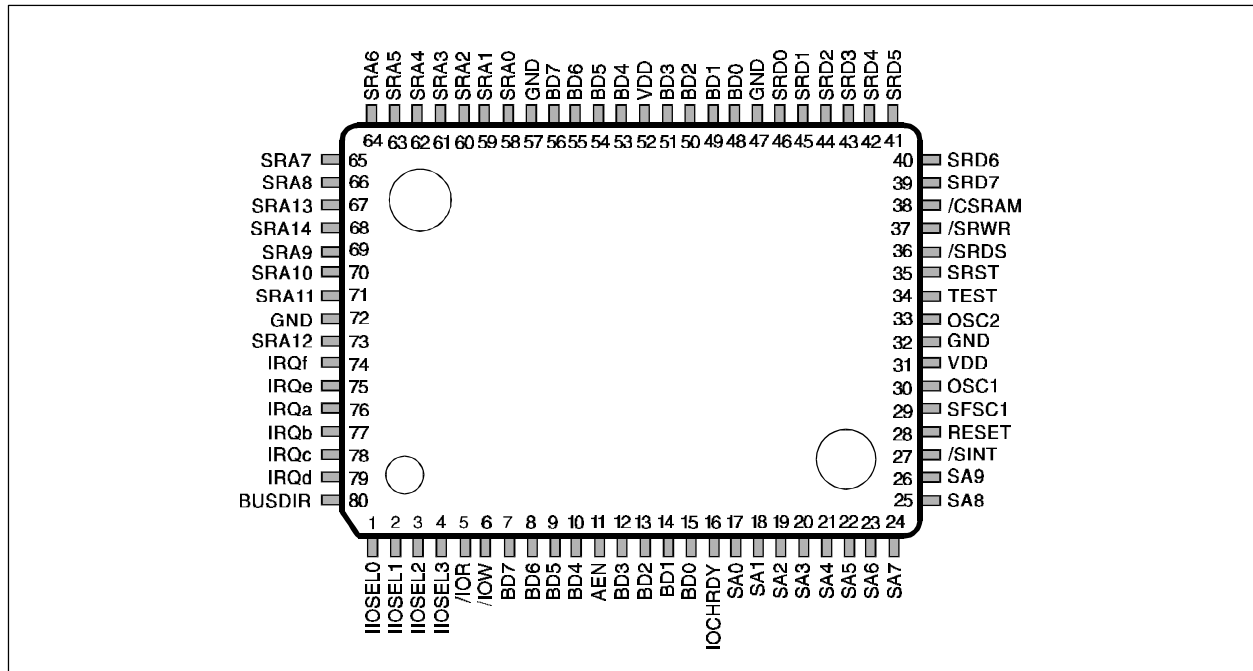


Figure 2: Pin Connection

2.1 ISA PC bus interface

Pin Name	Pin Number	Input (I) Output (O) Tristate-Out (OT)	Function
			ISA PC address bus
SA0	17	I	Address bit 0
SA1	18	I	Address bit 1
SA2	19	I	Address bit 2
SA3	20	I	Address bit 3
SA4	21	I	Address bit 4
SA5	22	I	Address bit 5
SA6	23	I	Address bit 6
SA7	24	I	Address bit 7
SA8	25	I	Address bit 8
SA9	26	I	Address bit 9
			ISA PC data bus
BD0	15, 48	I/O	Data bit 0 ¹⁾
BD1	14, 49	I/O	Data bit 1 ¹⁾
BD2	13, 50	I/O	Data bit 2 ¹⁾
BD3	12, 51	I/O	Data bit 3 ¹⁾
BD4	10, 53	I/O	Data bit 4 ¹⁾
BD5	9, 54	I/O	Data bit 5 ¹⁾
BD6	8, 55	I/O	Data bit 6 ¹⁾
BD7	7, 56	I/O	Data bit 7 ¹⁾

Pin Name	Pin Number	Input (I) Output (O) Tristate-Out (OT)	Function
/IOR	5	I	ISA PC bus control signals Read I/O enable
/IOW	6	I	Write I/O enable
AEN	11	I	Address enable
IOCHRDY	16	O	I/O Channel Ready
BUSDIR	80	O	Data Bus Direction
ISA PC Interrupt Requests			
IRQa	76	OT	Interrupt Request a
IRQb	77	OT	Interrupt Request b
IRQc	78	OT	Interrupt Request c
IRQd	79	OT	Interrupt Request d
IRQe	75	OT	Interrupt Request e
IRQf	74	OT	Interrupt Request f
Initial I/O Address Select			
IIOSEL0	1	I	Initial Address Select 0 ²⁾
IIOSEL1	2	I	Initial Address Select 1 ²⁾
IIOSEL2	3	I	Initial Address Select 2 ²⁾
IIOSEL3	4	I	Initial Address Select 3 ²⁾

2.2 S0 Device Interface

(for Address and Data bus see SRAM interface below)

/SINT	27	I	Interrupt from S0 device
SFSC1	29	I	Frame Sync 1 (neg. edge)
SRST	35	O	Reset to S0 device
/SRDS	36	O	Data Strobe to S0 device
/SRWR	37	O	Write enable to S0 device and SRAM

2.3 SRAM Interface

(SRA0 .. SRA5 are also input addresses of the S0 device)

SRAM address bus			
SRA0	58	O	Address bit 0
SRA1	59	O	Address bit 1
SRA2	60	O	Address bit 2
SRA3	61	O	Address bit 3
SRA4	62	O	Address bit 4
SRA5	63	O	Address bit 5
SRA6	64	O	Address bit 6
SRA7	65	O	Address bit 7
SRA8	66	O	Address bit 8
SRA9	69	O	Address bit 9

Pin Name	Pin Number	Input (I) Output (O) Tristate-Out (OT)	Function
SRA10	70	O	Address bit 10
SRA11	71	O	Address bit 11
SRA12	73	O	Address bit 12
SRA13	67	O	Address bit 13
SRA14	68	O	Address bit 14
SRAM and S0 device data bus			
SRD0	46	I/O	Data bit 0
SRD1	45	I/O	Data bit 1
SRD2	44	I/O	Data bit 2
SRD3	43	I/O	Data bit 3
SRD4	42	I/O	Data bit 4
SRD5	41	I/O	Data bit 5
SRD6	40	I/O	Data bit 6
SRD7	39	I/O	Data bit 7
/CSRAM	38	O	Chip Select SRAM
/SRWR	37	O	Write enable SRAM

2.4 Miscellaneous Pins

RESET	28	I	Master Reset ³⁾
OSC1	30	I	Quartz Oscillator or external clock input
OSC2	33	O	Quartz Oscillator
TEST	34	O	Test output, don't connect!

2.5 Power Pins

GND	32, 47, 57, 72		0 Volt Ground ⁴⁾
VDD	31, 52		+3 - +5 Volt ⁴⁾

¹⁾ Both pins must be connected to achieve 24 mA drive capability

²⁾ Pin has internal pull up resistor

³⁾ Pin has Schmitt Trigger input characteristic

⁴⁾ All Pins must be connected

3 Functional Description

3.1 Programming of ISA I/O-Addresses

The HFC always occupies two consecutive addresses in the I/O map of a PC. It decodes only the 10 lower address lines as most slot cards do on the ISA bus. On the lower of both addresses SA0 = 0; on the higher SA0=1.

After every Master Reset (RESET = 1) the I/O address select circuit inside the HFC is in hardware mode. So 16 different I/O addresses can be selected by the 4 inputs IIOSEL0 .. IIOSEL3 as Table. 1 shows below.

IIOSEL	Selected I/O address
3 2 1 0	
0 0 0 0	2F0h
0 0 0 1	2E0h
0 0 1 0	2D0h
0 0 1 1	210h
0 1 0 0	2C0h
0 1 0 1	200h
0 1 1 0	2F8h
0 1 1 1	2E8h
1 0 0 0	2B0h
1 0 0 1	3E0h
1 0 1 0	320h
1 0 1 1	278h
1 1 0 0	310h
1 1 0 1	330h
1 1 1 0	300h
1 1 1 1	3E8h

Table 1: Selected I/O address after reset

👉 important!

An open IIOSELx input has a logical value of 1 because of the internal pull up resistor.

The hardware selected I/O address might have an address collision with another I/O device in the PC.

After a hardware reset (RESET = 1) you must first write an I/O address into the HFC to set the I/O address for every further access to the device.

The procedure is as follows:

First you must write the lower 8 bits of the new I/O address you want into the lower address (SA0 = 0) of the hardware selected I/O address. The LSB of the new address is a don't care bit because the HFC always occupies two I/O addresses.

Then you have to write the additional 2 bits of the new I/O address into the higher address (SA0 = 1) of the hardware selected I/O address. The other 6 bits in the byte must have a special pattern to switch over to the software selected address mode. This pattern must be 0101 01aa, whereby aa are the 2 higher address bits.

e.g.: wanted I/O address: **3A4h / 3A5h**

IIOSEL(0:3) all 0s

then hardware selected I/O address is: **2F0h** = 10 1111 0000 b

write the value **A4h** or **A5h** into **2F0h** = 1010 010x b
 write the value **57h** into **2F1h** = 0101 01 11 b pattern address
 0101 0111 b

x = don't care

All further accesses to the HFC can only be done on the addresses **3A4h / 3A5h**. Only a master reset on the RESET pin will switch back the HFC into hardware selected address mode.

👉 hint:
 It's useful to solve a possible address conflict by programming the I/O address as early as possible. It is recommendable to set the address with a simple .SYS driver in a DOS environment.

3.2 ISA bus interface

The HFC only uses 2 I/O addresses with SA0 switches between data or control information. As normal only 10 bits of the ISA PC bus address are used for I/O address selection.

SA0	/IOR	/IOW	AEN	
X	X	X	1	no access
X	1	1	0	no access
0	0	1	0	read data
0	1	0	0	write data
1	0	1	0	read status
1	1	0	0	write control

The HFC has no memory or DMA access to any component on the ISA bus.

Because of its power drive characteristic it needs no external driver for the ISA bus data lines .

If necessary you can add an external bus driver. In this case the output BUSDIR determines the driver direction.

- BUSDIR = 1 means that data is driven into the HFC;
- BUSDIR = 0 means that the HFC is read and data is driven to the external bus.

3.3 Register Description

3.3.1 Control Register

3.3.1.1 Address Pointer (CAP)

Because the HFC uses only 2 I/O addresses an internal address pointer is used to select other registers.

address pointer	meaning
00ii iii	i: 6-Bit-address to S0 device (ISACS™) is copied to SRA0 .. SRA5 address lines

3.3.1.2 Internal pointer (CIP)

The internal pointer is used to select and control the FIFOs of the HFC.

10zz zzff	z: internal pointer	
	zzzz	
0000	FIFO input counter (Z1) low byte	r)
0001	FIFO input counter (Z1) high byte	r)
0010	FIFO output counter (Z2) low byte	r)
0011	FIFO output counter (Z2) high byte	r)
HDLC-Mode:		
1010	dummy for increm. of Frame counter (F1)	r)
1011	data write into FIFO, increment Z1	w)
1100	FIFO input HDLC Frame counter (F1)	r)
1101	FIFO output HDLC Frame counter (F2)	r)
1110	dummy for increm. of Frame counter (F2)	r)
1111	data read out of FIFO, increment Z2	r)
Transparent Mode:		
1010	data write into FIFO upside down and increment Z1	w)
1011	data write into FIFO, increment Z1	w)
1100	FIFO input HDLC Frame counter (F1)	r)
1101	FIFO output HDLC Frame counter (F2)	r)
1110	data read out of FIFO upside down and increment Z2	r)
1111	data read out of FIFO, increment Z2	r)

every other value for **zzzz** not allowed

r) corresponding data register is read only
w) corresponding data register is write only

f: FIFO-No.:			
00	channel B1 send	10	channel B2 send
01	channel B1 receive	11	channel B2 receive

👉 important!

If the FIFO-No. or the Frame counters (F1, F2) are changed it must be in a separate not BUSY period. That means BUSY must be active before the change and BUSY must be waited for after the change before any other function is initiated.
For BUSY see 3.3.2. ff.

3.3.1.3 Interrupt, Reset and Memory selects (CIRM)

110m rqqq

m: external memory select
m = 0 external RAM is 32k x 8
m = 1 external RAM is 8k x 8 the unused address bits are fixed to 1!

r: software reset Pin SRST goes high for external S0 device reset, all FIFO counters are initialized to all 1's, ISA I/O address is not changed, no control register is changed.

👉 important!

reset is active during "r = 1". It changes to inactive when "r = 0" is written into the register.

q:	IRQ-channel		
000	all deselected	100	IRQd
001	IRQa	101	IRQe
010	IRQb	110	IRQf
011	IRQc	111	all deselected

3.3.1.4 Transparent Mode and Timer controls (CTMT)

111c sett

c: clear timer interrupt
 this function is not latched as the software reset function above. Every write with "c = 1" resets the interrupt flip-flop but doesn't block it for further interrupts.

s: select timer period
s = 0 timer period is 25 ms
s = 1 timer period is 50 ms

e: timer interrupt enable 1 = enable 0 = disable

channel: B2 B1
tt: t₁ t₀ transparent mode if t-Bit set (see 3.6.3.)

3.3.1.5 Control Register Initialisation

All control registers are reset to 0 when RESET-Pin is active.

3.3.2 Status Register

There is only one status register. It is readable with (SA0 = 1).

xxxx xteb **t**: timer interrupt pending
this bit is set (=1) if the time period of the timer has run down. It can be reset by the c-bit in the CTMT register (3.3.1.4.)

e: external interrupt pending
this bit is an inverted copy of the /SINT input

b: BUSY
this bit is set (=1) if the device is internally busy and cannot be accessed from outside. You must not initiate any access via the data read or write port when BUSY is active. Independent of BUSY the control registers can be written into and the status can be read at any time.

x: unknown
these bits are undefined when status is read. You must ignore them.

3.3.3 Data read and write

Data can be read or written with SA0 = 0. This is the data port. Because of internal activity and external SRAM access the HFC enters a BUSY period after each high-low transition on the SFSC1 input. This BUSY period has a duration of app. 350 clock transitions.



important!

During BUSY there must not be any access on the data port (SA0 = 0).

To synchronise the data accesses you can use two methods.

3.3.3.1 Software BUSY synchronisation

Before any data access you must read the status port. If BUSY bit is active (= 1) you repeat reading the status port until BUSY changes to inactive. If BUSY is inactive (= 0) you can access the data port.

Pseudo code:

```
10  Read BUSY bit
    if BUSY goto 10
    read or write data
```



important!

To ensure proper BUSY synchronisation start of BUSY and internal functions are inhibited after a status read with BUSY inactive until a data port access is done.

So you must guarantee that your software driver immediately accesses the data port after any status read with BUSY inactive (= 0). The time delay between status read and data port access should be as small as possible. So all interrupts should be disabled during this time intervall.

3.3.3.2 Hardware BUSY synchronisation

If the output named IOCHRDY is connected to a processor /WAIT line the data port is accessible without any status read. If the HFC is BUSY the IOCHRDY line changes to low and the PC processor waits until BUSY is inactive. Because of the fast internal processing of the HFC the IOCHRDY line is low for a maximum of 450 clock transitions. So the maximum waiting period is limited to an amount where no timing problems with refresh of dynamic RAMs or interrupt problems are to be considered.

3.4 Interrupt

There are two possible interrupt sources:

1. internal Timer
2. external Interrupt from S0 device on /SINT

Both interrupt sources are "ored" and the result is routed to one of six interrupt outputs. There is no more than one active interrupt output at any time. The choice of the interrupt line can be taken in the CIRM control register.

To check which one of the interrupt sources has a pending interrupt the interrupt status can be read with the status read.

 **important!**

Remember if you read the status port with BUSY not active you must access the data port immediately afterwards.

3.5 Timer

The HFC internal timer is based on a time period of 25ms or 50ms. The timer clock is not derived from the clock input but from the SFSC1 signal. This signal must be fed with a clock signal of 125µs period which normally comes from the S0 device.

The timer interrupt is maskable, but the status bit always reflects the state of the interrupt flip-flop. There is no possibility to reset the timer. So every timer interrupt comes exactly after the selected time period independent of the reset time of the interrupt flip-flop.

3.6 FIFOs

There are 4 FIFOs with 4 HDLC-Controllers in the HFC. The HDLC circuits are located on the S0 device side of the HFC. So plain data is always stored in the FIFO. Zero insertion and deletion is only done:

- if the data goes to the S0 device in send FIFOs and
- when the HDLC data comes from the S0 device in receive operation.

There are a send and a receive FIFO for each of the two B-channels.

The FIFOs are realized as ring buffers in the external SRAM. To control them there are some counters.

Z1: FIFO input counter (13 Bit)

Z2: FIFO output counter (13 Bit)

Each counter points to a byte position in the SRAM. On a FIFO input operation Z1 is automatically incremented. On an output operation Z2 is incremented.

After every high-low transition on the SFSC1 input signal two HDLC-bytes are written into the S0 device (FIFOs No. 0 and 2) and two HDLC-bytes are read from the S0 device (FIFOs No. 1 and 3).

If $Z1 = Z2$ the FIFO is empty.

Additionally there are two counters F1 and F2 for every FIFO channel (5Bit). They count the HDLC-frames in the FIFOs and form a ring buffer as Z1 and Z2 do, too.

Again F1 is incremented when a complete frame has been received and stored in the FIFO. F2 is incremented when a complete frame has been read from the FIFO.

If $F1 = F2$ there is no complete frame in the FIFO.

When the RESET line is active or software reset is active Z1, Z2, F1 and F2 are all initialized to all 1s. The F-counters are true ring counters. So after an increment of 1Fh the counter state is 00 h.

 **important!**

The counter state 0200h of the Z-counters follows counter state 1FFFh.

3.6.1 FIFO channel operation

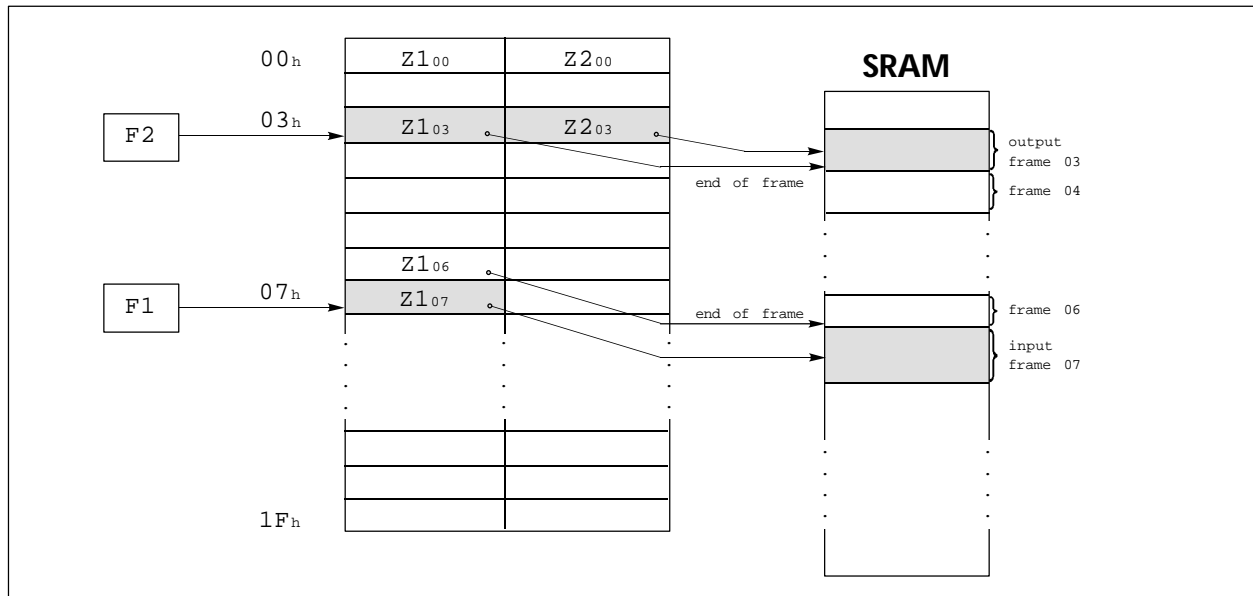


Figure 3: FIFO-Organisation

3.6.1.1 Send Channel (FIFO-No. 0 and 2)

The send channels send data from the ISA-PC bus interface to the FIFO and the HFC converts the data into HDLC code and transfers it from the FIFO into the S0 device write registers.

The HFC check Z1 and Z2. If $Z1=Z2$ (FIFO empty) the HFC generates a HDLC-Flag (0111 1110) and sends it to the S0 device. In this case Z2 is not incremented. If also $F1=F2$ only HDLC flags are send to the S0 device and all counters remain unchanged. If the frame counters are unequal F2 is incremented and the HFC tries to send the next frame to the S0 device. After the end of a frame (Z2 reaches Z1) it automatically generates the 16 bit CRC checksum and adds the ending flag. If there is another frame in the FIFO the ending flag of the last frame is also the starting flag of the next frame.

With every byte you send to the FIFO via the ISA-PC bus interface Z1 is incremented automatically. If a complete frame has been send F1 must be incremented to send the next frame. If the frame counter F1 is incremented also the Z-counters may change because Z1 and Z2 are functions of F1 and F2. So there are $Z1(F1)$, $Z2(F1)$, $Z1(F2)$ and $Z2(F2)$.

$Z1(F1)$ is used for the frame which is just received from the PC-bus side. $Z2(F2)$ is used for the frame which is just transmitted to the S0 device side of the HFC. $Z1(F2)$ is the end of frame pointer of the current output frame.

In the send channels F1 is only changed from the PC interface side if the software driver wants to say „end of send frame“. Then the current value of Z1 is stored, F1 and Z1 are incremented and Z1 is used as start address of the next frame.

important!

At the start of the first frame when the FIFO is totally empty you must put at least two bytes into the FIFO before a BUSY condition is initialized by the HFC. This is necessary to avoid the initialisation of a CRC sequence after a one-byte frame. To satisfy this condition you should wait for a BUSY / no BUSY status transition. In this case there is enough time to write more than one byte into the FIFO.

3.6.1.2 FIFO full condition in send channels

Due to the limited number of registers in the HFC the driver software must maintain a list of frame start and end addresses to calculate actual FIFO depth and check FIFO full condition. Because there are a maximum of 32 frame counter values and the start address of a frame is the incremented value of the frame end address the memory table must have only 32 values of 16 bits (13 bits) instead of 64.

Remember that an increment of Z-value 1FFFh is 0200h!

There are two different FIFO full conditions. The first one is met when the FIFO contents comes up to 31 frames. There is no possibility for the HFC to manage more than 31 frames even if the frames are very small.

The second limitation is the depth of the FIFO which is 1.5 Kbyte with a 8 Kbyte external SRAM and 7.5 Kbyte with a 32 Kbyte external SRAM.

3.6.1.3 Receive Channels (FIFO-No. 1 and 3)

The receive channels receive data from the S0 device read registers. The data is converted from HDLC into plain data and send to the FIFO. The data can then be read via the ISA-PC bus interface.

The HFC checks the HDLC data coming in. If it finds a flag or more than 6 consecutive 1s it does not generate any output data. In this case Z1 is not incremented. Proper HDLC-data being received is converted by the HFC into plain data. After the ending flag the HFC checks the HDLC CRC checksum. If it is correct one byte with all 0s is inserted behind the CRC data in the FIFO. This last byte of a frame in the FIFO is different from all 0s if there is no correct CRC field at the end of the frame.

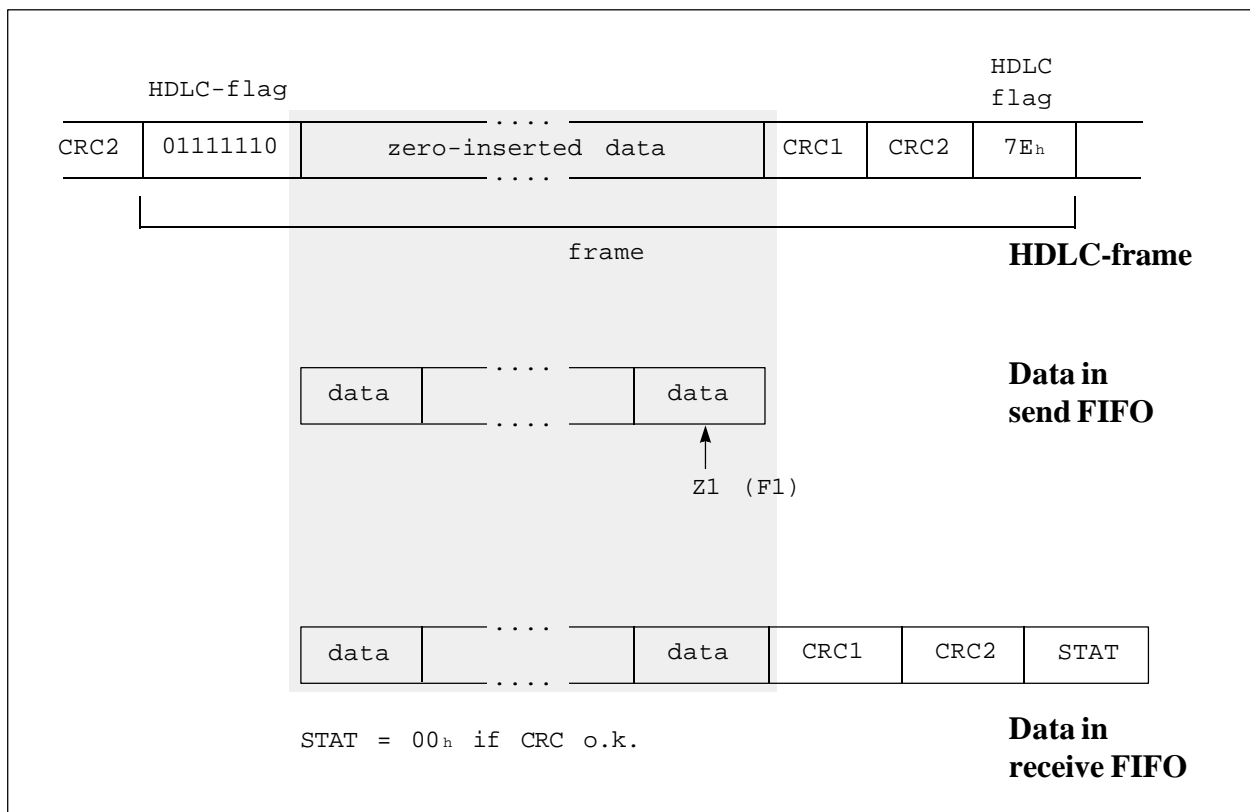


Figure 4: FIFO Data Organisation

The ending flag of a HDLC-frame can also be the starting flag of the next frame.

After a frame is received completely F1 is incremented by the HFC automatically and the next frame starts.

After reading a frame via the ISA-PC bus interface F2 must be incremented. If the frame counter F2 is incremented also the Z-counters may change because Z1 and Z2 are functions of F1 and F2. So there are Z1(F1), Z2(F1), Z1(F2) and Z2(F2).

Z1(F1) is used for the frame which is just received from the S0 device side of the HFC. Z2(F2) is used for the frame which is just transmitted to the ISA-PC bus interface. Z1(F2) is the end of frame pointer of the current output frame.

To calculate the length of the current receive frame the software has to evaluate Z1-Z2. When Z2 reaches Z1 the complete frame has been read.

In the receive channels F2 must be incremented from the PC interface side after the software detects an end of receive frame ($Z1=Z2$) and $F1 \neq F2$. Then the current value of Z2 is stored, F2 and Z2 are incremented and Z2 is copied as start address of the next frame.

If $Z1 = Z2$ and $F1 = F2$ the FIFO is totally empty.

3.6.1.4 FIFO full condition in receive channels

Because the ISDN-B-channels have no hardware-based flow control there is no possibility to stop input data if a receive FIFO is full.

So there is no FIFO full condition implemented in the HFC. The HFC assumes that the PC processor hardware is able to avoid any overflow of the receive FIFOs. Overflow conditions are again more than 31 input frames or a real overflow of the FIFO because of excessive data.

Because HDLC procedures only know a window size of 7 frames no more than 7 frames are send without software intervention. Due to the great depth of the FIFOs of the HFC it is easy to poll the HFC even in large time intervalls without having to fear a FIFO overflow condition.

However to avoid any undetected FIFO overflows the software driver should check the number of frames in the FIFO which is F1-F2. An overflow exists if the number (F1-F2) is less than the number in the last reading even if there was no reading of a frame in between.

After a detected FIFO full condition the HFC must be reset via the software or hardware RESET!

3.6.2 FIFO initialisation

All counters Z1, Z2, F1 and F2 of all FIFOs are initialized to all 1s after a RESET. The RESET signal must have a length of at least 4 clock cycles.

Then the result is $Z1 = Z2 = 1FFF_h$ and $F1 = F2 = 1F_h$.

The same initialisation is done if the r-bit in the CIRM register is set.

3.6.3 Transparent mode of HFC

You can switch off HDLC operation for each B-channel. There is one t-bit for each B-channel in the CTMT control register. If this bit is set data in the FIFO is send directly to the S0 device and data from the S0 device is send directly to the FIFO.

Be sure to switch into transparent mode only if F1=F2. Being in transparent mode the Fx counters remain unchanged. Z1 and Z2 are the input and output pointers respectively. Because F1=F2 both Z-counters are always accessible.

If a send FIFO channel changes to FIFO empty condition no CRC is generated and the last data byte is repeated until there is new data.

In receive channels there is no check on flags or correct CRCs and no status byte is added.

The byte boundaries are not arbitrary like in HDLC mode where byte synchronisation is achieved with HDLC-flags. The data is just the same as it comes from the S0 device or is send to this.

Because Fx incrementation dummy registers are not used you can send and receive transparent data in two shapes. The normal and first shape is tranporting B-channel data with the LSB first as it is in HDLC mode. The second shape is sending the bytes upside down as it is normal for PWM data. So the first bit is the MSB.

3.7 External SRAM

For the FIFO data an external SRAM is used. Two sizes of 8 Bit orientated SRAMs can be used:

- 8K x 8 for 4 x 1.5 Kbyte FIFO depth or
- 32K x 8 for 4 x 7.5 Kbyte FIFO depth.

The first 2 Kbyte of the SRAM are reserved for internal HFC use.

👉 hint!

If you connect the HFC with the SRAM you can simplify PCB layout if you permutate address lines and data lines. If you connect data lines of the SRAM with data lines of the HFC and SR-address lines of the HFC with address lines of the SRAM you can do this in any order.

3.8 S0 device

The HFC is designed to interface directly with an ISACS™ of SIEMENS AG Germany.

It uses the following ISACS™ internal registers to read and write B-channel data.

ISACS™ register name	address	FIFO-No.	HFC read/write
C1R	35h	0	write
B1CR	37h	1	read
C2R	36h	2	write
B2CR	38h	3	read

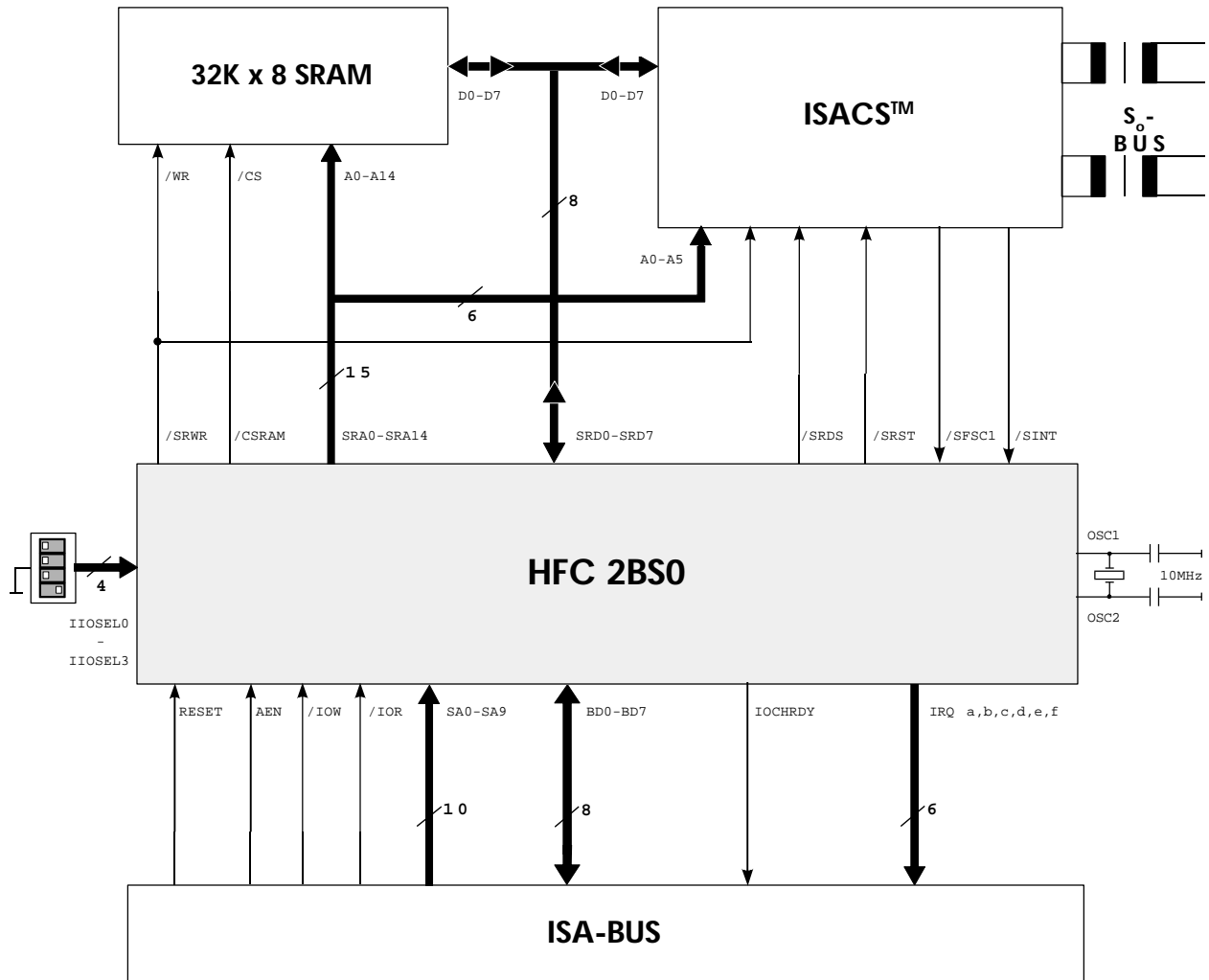


Figure 5: Typical Application

4. Electrical Characteristics

Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply voltage	V_{cc}	-0.3 to + 7.0	V
Input voltage	V_I	-0.3 to $V_{cc} + 0.3$	
Output voltage	V_O	-0.3 to $V_{cc} + 0.3$	
Operating temperature	T_{opr}	-10 to + 70	°C
Storage temperature	T_{stg}	-55 to + 150	

Recommended Operating Conditions for TTL and CMOS interface

Parameter	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Supply voltage	V_{cc}		3.0	5.0	5.25	V
Supply current	I_{CC}	$f_{CLK}=10MHz$		8		mA
Operating temperature	T_{opr}		0		+70	°C

Electrical Characteristics

$V_{cc} = 4.75$ to $5.25V$ (TTL), $V_{cc} = 4.5$ to $5.5V$ (CMOS), $T_{opr} = -10$ to $70^\circ C$

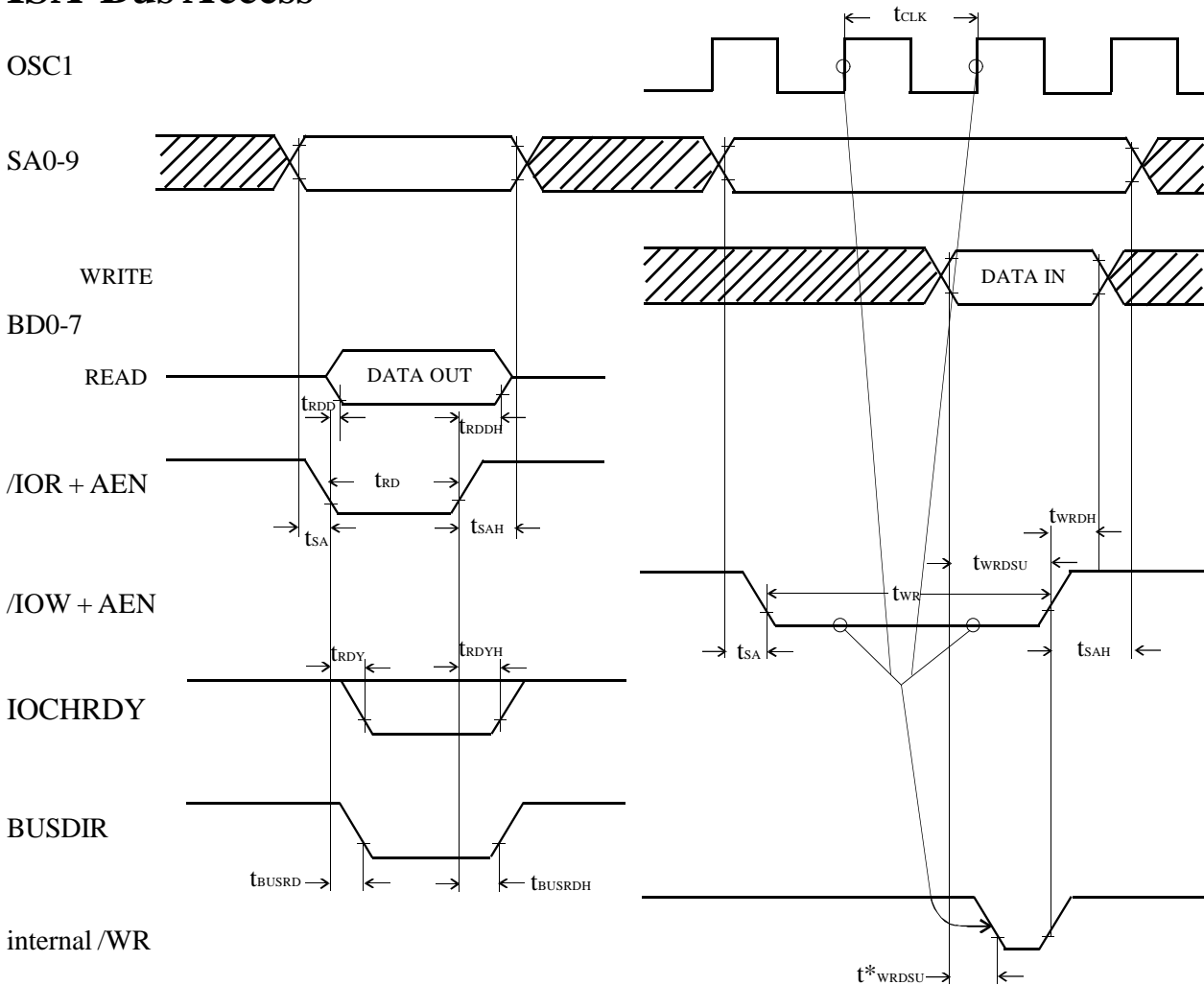
Parameter	Symbol	Condition	TTL level			CMOS level			Unit
			MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Input LOW voltage	V_{IL}				0.8			1.5	V
Input HIGH voltage	V_{IH}		2.0			3.5			
Input HIGH threshold voltage	V_{T+}	Schmitt input buffer			2.2			3.7	
Input LOW threshold voltage	V_{T-}		0.5			1.0			
Hysteresis voltage	V_H		0.2			0.4			
Output LOW voltage	V_{OL}		$I_{OL} = 4mA$			0.4			
Output HIGH voltage	V_{OH}	$I_{OH} = -2mA$	4.0			4.0			
Output leakage current	$ I_{OZ} $	High Z			10			10	µA
Pull-up resistor input current	$ I_{IL} $	$V_I = 0V$	8.0		60	8.0		60	

I/O Characteristics

Input	Interface Level
SA0-9	TTL
BD0-7	TTL
/IOR	TTL
/IOW	TTL
IIOSEL0-3	TTL with Pull Up
/SINT	TTL
SFSC1	TTL
SRD0-7	TTL
RESET	TTL Schmitt Trigger

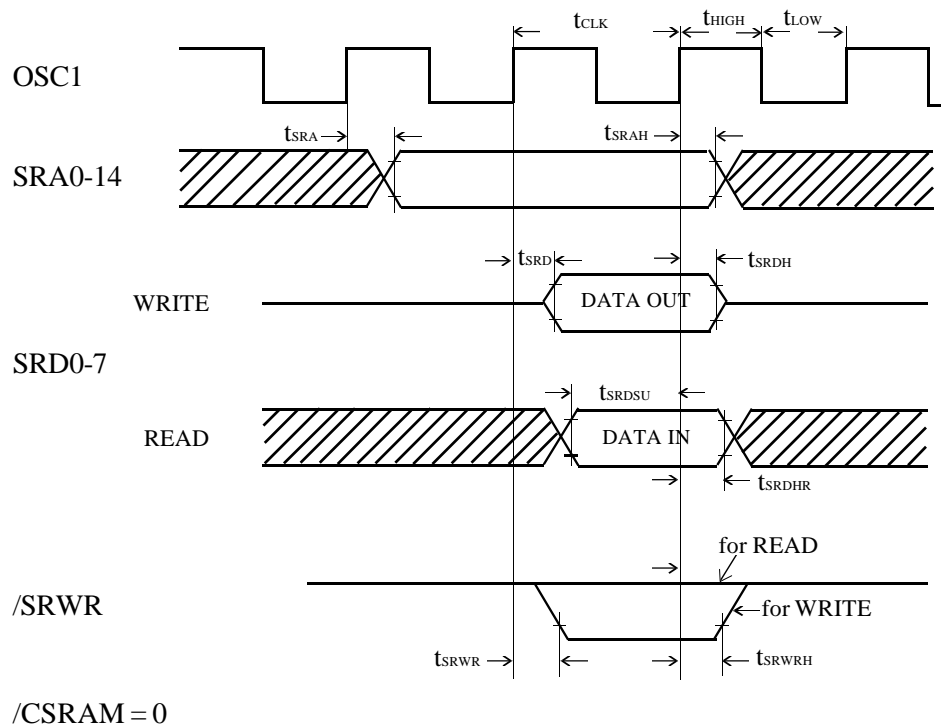
Output	Driver Capability		
	Low		High
	0,4V	0,6V	
BD0-7	16mA	24mA	8mA
IOCHRDY	6mA		3mA
BUSDIR	4mA		2mA
IRQa-f	6mA		3mA
SRST	4mA		2mA
/SRDS	4mA		2mA
/SRWR	4mA		2mA
SRA0-14	4mA		2mA
SRD0-7	4mA		2mA
\CSRAM	4mA		2mA

ISA-Bus Access



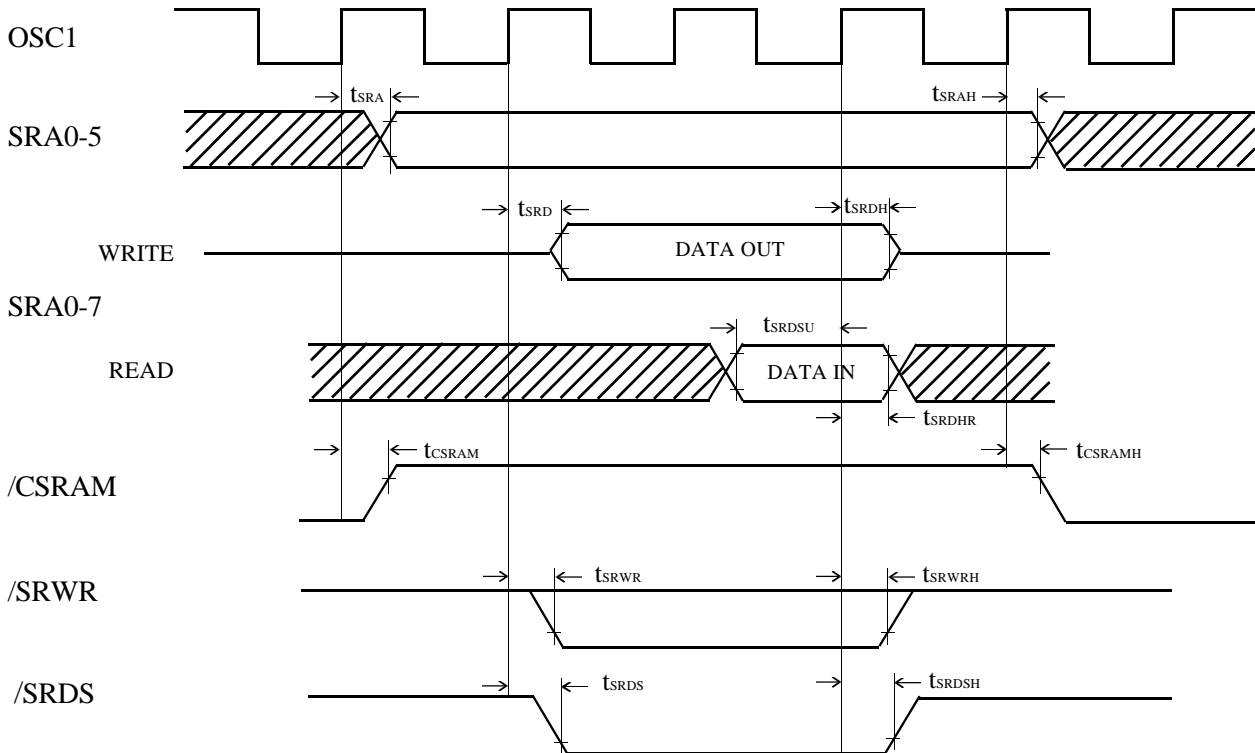
SYMBOL	CHARACTERISTICS	MIN.	MAX.
t_{RDD}	/IOR Low to Data Out Time	3ns	25ns
t_{RDDH}	/IOR High to Data Out Hold Time	2ns	15ns
t_{SA}	Address to /IOR or /IOW Low Setup Time	5ns	—
t_{SAH}	Address Hold Time after /IOR or /IOW High	5ns	—
t_{RD}	Read Time	30ns	∞
t_{WR}	Write Time	30ns	∞
	Write Time for Control Register WRITE	$2 \times t_{clk} + 20ns$	∞
t_{WRDSU}	Data Setup Time to /IOW Low	30ns	∞
t^*_{WRDSU}	Data Setup Time to internal /WR Low for Control Register WRITE	30ns	∞
t_{WRDH}	Data Hold Time from /IOW High	10ns	—
t_{RDY}	Delay Time from /IOR or /IOW Low to IOCHRDY Low	3ns	30ns
t_{RDYH}	Delay Time from /IOR Low or /IOW High to IOCHRDY High	3ns	30ns
t_{BUSRD}	Delay Time from /IOR Low to BUSDIR Low	3ns	25ns
t_{BUSRDH}	Delay Time from /IOR High to BUSDIR High	2ns	15ns

SRAM Access



SYMBOL	CHARACTERISTICS	MIN.	MAX.
t_{LOW}	Clock Low Level Width	25ns	∞
t_{HIGH}	Clock High Level Width	25ns	∞
t_{CLK}	Clock Cycle Time	50ns	∞
t_{SRA}	Address Stable after Clock \uparrow	20ns	80ns
t_{SAH}	Address Stable Hold Time after Clock \uparrow	10ns	–
t_{SRD}	Data Out Stable after Clock \uparrow	15ns	60ns
t_{SRDH}	Data Out Stable Hold Time after Clock \uparrow	10ns	–
t_{SRDSU}	Data In Setup Time to Clock \uparrow	15ns	–
t_{SRDHR}	Data In Hold Time after Clock \uparrow	10ns	–
t_{SRWR}	Delay Time Clock \uparrow to /SRWR Low	2ns	40ns
t_{SRWRH}	Delay Time Clock \uparrow to /SRWR High	5ns	40ns

S0 Device Access



SYMBOL	CHARACTERISTICS	MIN.	MAX.
t_{SRA}	Address Stable Delay after Clock \uparrow	20ns	80ns
t_{SRAH}	Address Stable Hold Time after Clock \uparrow	10ns	–
t_{SRD}	Data Out Stable after Clock \uparrow	15ns	60ns
t_{SRDH}	Data Out Stable Hold Time after Clock \uparrow	10ns	–
t_{SRDSU}	Data In Setup Time to Clock \uparrow	15ns	–
t_{SRDHR}	Data In Hold Time after Clock \uparrow	10ns	–
t_{CSRAM}	Delay Time Clock \uparrow to /CSRAM High	10ns	60ns
t_{CSRAMH}	Hold Time Clock \uparrow to /CSRAM Low	10ns	–
t_{SRWR}	Delay Time Clock \uparrow to /SRWR Low	2ns	40ns
t_{SRWRH}	Delay Time Clock \uparrow to /SRWR High	5ns	40ns
t_{SRDS}	Delay Time Clock \uparrow to /SRDS Low	2ns	40ns
t_{SRDSH}	Hold Time Clock \uparrow to /SRDS High	5ns	40ns



**Eintrachtstrasse 113
D-50668 Köln
Germany**

**Tel.: +49 (0) 22 1 / 13 67 35
Fax: +49 (0) 22 1 / 13 47 15**