Data Sheet

Cologne Chip

HFC - S active

ISDN Microprocessor

(ARM7 based)

Preliminary Data Sheet: July 2002

Cologne Chip AG
Eintrachtstrasse 113
D - 50668 Köln
Germany

Tel.: +49 (0) 221 / 91 24-0
Fax: +49 (0) 221 / 91 24-100

http://www.CologneChip.com
http://www.CologneChip.de
support@CologneChip.com

Contents

General Remarks to Notations

1. Numerical values have different notations for various number systems, e.g. the hexadecimal value 0xC9 is binary '11001001' and in decimal notation 201.
2. The first letter of registers and their bit (resp. bitmap) names indicates the typ: 'R_ . . . ' is a register, 'A_ . . . ' is an array-register, 'V_ . . . ' is a bit or bitmap value and 'M_ . . . ' is its bitmap mask, i.e. all bits of the bitmap are set to '1'.

List of Figures

List of Tables

List of Registers (sorted by name)

The first letter of the register names indicates the typ: ‘R_ . . . ’ is a register, ‘A_ . . . ’ is an array-register. The index of array-registers is either the FIFO, channel or slot which has to be specified in the appropriate register.

Address	Width	Name	Mode	Page
0x00080040	16	R_CNT1B_CFG	r/w	46
0x000D000C	32	R_CODEC_CTRL	r/w	125
0x000B01E0	32	R_CODEC_IDX	r/w	121
0x000D0004	32	R_CODEC_RX	r	123
0x000D0008	32	R_CODEC_RX8	r	124
0x000D0000	32	R_CODEC_TX	r/w	123
0x00080028	32	R_DIV1_CFG	r/w	45
0x00080000	16	R_FIQ_CTRL	r/w	57
0x00080008	16	R_FIQ_STATUS	r/w	59
0x00090030	32	R_FSC_CFG	r/w	67
0x00090034	32	R_FSC_CONST	r/w	68
0x00090028	16	R_FSC_IRQ	r/w	66
0x00090014	32	R_GPIO_CFG	r/w	129
0x00090024	32	R_GPIO_CTRL1	r/w	131
0x00080030	32	R_GPIO_CTRL2		133
0x00090020	32	R_GPIO_IN1	r	130
0x00080034	16	R_GPIO_IN2	r	130
0x00090018	32	R_GPIO_IRQ_CTRL	r/w	129
0x0009001C	32	R_GPIO_OUT	r/w	130
0x0008002C	32	R_GPO_CTRL	r/w	132
0x000B0210	32	R_HW_SL_CNT	r	116
0x000B0204	16	R_HW1_CTRL	r/w	111
0x000B00C0	32	R_HW1_IDX	r/w	120
0x000B0120	32	R_HW1_RX_CUR	r/w	105
0x000B0020	32	R_HW1_RX_LAST	r/w	103
0x000B01E8	32	R_HW1_TS_EN	r/w	107
0x000B0100	32	R_HW1_TX_CUR	r/w	105
0x000B0000	32	R_HW1_TX_NEXT	r/w	103
0x000B0208	16	R_HW2_CTRL	r/w	113
0x000B00E0	32	R_HW2_IDX	r/w	120
0x000B0160	32	R_HW2_RX_CUR	r/w	106
0x000B0060	32	R_HW2_RX_LAST	r/w	104
0x000B01EC	32	R_HW2_TS_EN	r/w	107
0x000B0140	32	R_HW2_TX_CUR	r/w	106
0x000B0040	32	R_HW2_TX_NEXT	r/w	104
0x000B020C	16	R_HW3_CTRL	r/w	115
0x000B01C0	32	R_HW3_IDX	r/w	120
0x000B01A0	32	R_HW3_RX_CUR	r/w	107
0x000B00A0	32	R_HW3_RX_LAST	r/w	105
0x000B01F0	32	R_HW3_TS_EN	r/w	108
0x000B0180	32	R_HW3_TX_CUR	r/w	106
0x000B0080	32	R_HW3_TX_NEXT	r/w	104
0x00080004	16	R_IRQ_CTRL	r/w	58
0x0008000C	16	R_IRQ_STATUS	r/w	60
0x00080038	8	R_OSC_CFG	r/w	45
0x000B0200	8	R_PCM_CFG	r/w	110
0x000B01F4	16	R_PFS0_CFG	r/w	108

Address	Width	Name	Mode	Page
0x000B01F6	16	R_PFS1_CFG	r/w	108
0x000B01F8	16	R_PFS2_CFG	r/w	109
0x000B01FA	16	R_PFS3_CFG	r/w	109
0x00080020	32	R_PLL1_CFG	r/w	44
0x00080024	32	R_PLL2_CFG	r/w	44
0x0009000C	32	R_PWM_CFG	r/w	52
0x00080010	16	R_SDRAM_CTRL	r/w, r	36
0x000C0038	16	R_ST_B1_CRC	r/w	81
0x000C0028	32	R_ST_B1_RX_FIFO	r/w	80
0x000C00F0	8	R_ST_B1_RX	r	90
0x000C001C	32	R_ST_B1_TX_FIFO	r/w	79
0x000C00F4	8	R_ST_B1_TX	w	90
0x000C0014	32	R_ST_B12_IRQ_EN	r/w	78
0x000C000C	32	R_ST_B12_IRQ_STATUS	r/w	77
0x000C0048	32	R_ST_B12_STATUS	r	84
0x000C003C	16	R_ST_B2_CRC	r/w	82
0x000C0030	32	R_ST_B2_RX_FIFO	r/w	81
0x000C00F8	8	R_ST_B2_RX	r	90
0x000C0020	32	R_ST_B2_TX_FIFO	r/w	79
0x000C00FC	8	R_ST_B2_TX	w	90
0x000C0000	32	R_ST_CFG	r/w	72
0x000C00DC	8	R_ST_CLK_CTRL	w	89
0x000C0044	8	R_ST_CTRL	r/w	83
0x000C00C4	8	R_ST_CTRL1	w	87
0x000C00C8	8	R_ST_CTRL2	w	88
0x000C00CC	8	R_ST_CTRL3	w	88
0x000C0040	16	R_ST_D_CRC	r/w	82
0x000C0010	16	R_ST_D_FIFO_STATUS	r/w	78
0x000C0018	16	R_ST_D_IRQ_EN	r/w	78
0x000C0034	32	R_ST_D_RX_FIFO	r/w	81
0x000C0100	8	R_ST_D_RX	r	91
0x000C004C	16	R_ST_D_STATUS	r	84
0x000C0024	32	R_ST_D_TX_FIFO	r/w	80
0x000C0104	8	R_ST_D_TX	w	91
0x000C0108	8	R_ST_E_RX	r	91
0x000B01E4	16	R_ST_IDX	r/w	121
0x000C00C0	8	R_ST_RD_STATES	r	85
0x000C0008	32	R_ST_RX_STATUS	r	74
0x000C00D0	8	R_ST_SQ_MF	r/w, r, w	89
0x000C0004	32	R_ST_TX_STATUS	r	74
0x000C00C0	8	R_ST_WR_STATES	w	86
0x00090004	32	R_TIMER_CFG1	r/w	51
0x00090010	32	R_TIMER_CFG2	r/w	53
0x0009002C	32	R_TIMER_PRELD	r/w	50
0x00090000	32	R_TIMER	r/w	50
0x000A0010	16	R_UART_BAUD	w	138
0x000A0020	32	R_UART_CFG	w	139
0x000A0024	8	R_UART_CLR	w	140
0x000A0028	8	R_UART_ECHO	r/w	140
0x000A002C	32	R_UART_IRQ_CFG	r/w	144
0x000A0020	32	R_UART_PREVIEW	r	142
0x000A0000	32	R_UART_RX1	r	140
0x000A0004	32	R_UART_RX2	r	141
0x000A0008	32	R_UART_RX3	r	141
0x000A000C	32	R_UART_RX4	r	142

Address	Width	Name	Mode	Page
0x000A0024	32	R_UART_STATUS	r	143
0x000A0000	16	R_UART_TX1	w	137
0x000A0004	16	R_UART_TX2	w	137
0x000A0008	16	R_UART_TX3	w	137
0x000A000C	16	R_UART_TX4	w	138
0x000E0000	8	R_USB_ADDR	r/w	147
0x000E0004	8	R_USB_CFG	r/w, r	148
0x000E0008	8	R_USB_CTRL	r/w	148
0x0008003C	8	R_USB_DRV	r/w, r	147
0x000E000C	8	R_USB_EV1	r	149
0x000E0014	8	R_USB_EV2	r	150
0x000E0010	8	R_USB_EVMSK1	r/w	150
0x000E0018	8	R_USB_EVMSK2	r/w	151
0x000E0038	8	R_USB_JCMD	r/w	154
0x000E0034	8	R_USB_JDATA	w	153
0x000E0044	8	R_USB_JEP_EN	r/w, r	155
0x000E0058	32	R_USB_JEP_EV	r/w	157
0x000E005C	32	R_USB_JEP_EVMSK	r/w	158
0x000E0030	8	R_USB_JEP_SEL	r/w	153
0x000E004C	8	R_USB_JEP_STALL	r/w	156
0x000E003C	8	R_USB_JSTATUS	r	154
0x000E0028	8	R_USB_OCMD	r/w	152
0x000E0024	8	R_USB_ODATA	r	151
0x000E0040	8	R_USB_OEP_EN	r/w, r	155
0x000E0050	8	R_USB_OEP_EV	r	157
0x000E0054	32	R_USB_OEP_EVMSK	r/w	157
0x000E0020	8	R_USB_OEP_SEL	r/w	151
0x000E0048	8	R_USB_OEP_STALL	r/w	156
0x000E002C	8	R_USB_OSTATUS	r	152
0x00090008	32	R_WD	r/w	51
0x00080018	32	R_WS1	r/w	37
0x0008001C	32	R_WS2	r/w	38

List of Registers (sorted by address)

The first letter of the register names indicates the typ: ‘R_ . . . ’ is a register, ‘A_ . . . ’ is an array-register. The index of array-registers is either the FIFO, channel or slot which has to be specified in the appropriate register.

Address	Width	Name	Mode	Page
0x00080000	16	R_FIQ_CTRL	r/w	57
0x00080004	16	R_IRQ_CTRL	r/w	58
0x00080008	16	R_FIQ_STATUS	r/w	59
0x0008000C	16	R_IRQ_STATUS	r/w	60
0x00080010	16	R_SDRAM_CTRL	r/w, r	36
0x00080018	32	R_WS1	r/w	37
0x0008001C	32	R_WS2	r/w	38
0x00080020	32	R_PLL1_CFG	r/w	44
0x00080024	32	R_PLL2_CFG	r/w	44
0x00080028	32	R_DIV1_CFG	r/w	45
0x0008002C	32	R_GPO_CTRL	r/w	132
0x00080030	32	R_GPIO_CTRL2		133
0x00080034	16	R_GPIO_IN2	r	130
0x00080038	8	R_OSC_CFG	r/w	45
0x0008003C	8	R_USB_DRV	r/w, r	147
0x00080040	16	R_CNT1B_CFG	r/w	46
0x00090000	32	R_TIMER	r/w	50
0x00090004	32	R_TIMER_CFG1	r/w	51
0x00090008	32	R_WD	r/w	51
0x0009000C	32	R_PWM_CFG	r/w	52
0x00090010	32	R_TIMER_CFG2	r/w	53
0x00090014	32	R_GPIO_CFG	r/w	129
0x00090018	32	R_GPIO_IRQ_CTRL	r/w	129
0x0009001C	32	R_GPIO_OUT	r/w	130
0x00090020	32	R_GPIO_IN1	r	130
0x00090024	32	R_GPIO_CTRL1	r/w	131
0x00090028	16	R_FSC_IRQ	r/w	66
0x0009002C	32	R_TIMER_PRELD	r/w	50
0x00090030	32	R_FSC_CFG	r/w	67
0x00090034	32	R_FSC_CONST	r/w	68
0x000A0000	32	R_UART_RX1	r	140
0x000A0000	16	R_UART_TX1	w	137
0x000A0004	32	R_UART_RX2	r	141
0x000A0004	16	R_UART_TX2	w	137
0x000A0008	32	R_UART_RX3	r	141
0x000A0008	16	R_UART_TX3	w	137
0x000A000C	32	R_UART_RX4	r	142
0x000A000C	16	R_UART_TX4	w	138
0x000A0010	16	R_UART_BAUD	w	138
0x000A0020	32	R_UART_PREVIEW	r	142
0x000A0020	32	R_UART_CFG	w	139
0x000A0024	32	R_UART_STATUS	r	143
0x000A0024	8	R_UART_CLR	w	140
0x000A0028	8	R_UART_ECHO	r/w	140
0x000A002C	32	R_UART_IRQ_CFG	r/w	144
0x000B0000	32	R_HW1_TX_NEXT	r/w	103
0x000B0020	32	R_HW1_RX_LAST	r/w	103

Address	Width	Name	Mode	Page
0x000B0040	32	R_HW2_TX_NEXT	r/w	104
0x000B0060	32	R_HW2_RX_LAST	r/w	104
0x000B0080	32	R_HW3_TX_NEXT	r/w	104
0x000B00A0	32	R_HW3_RX_LAST	r/w	105
0x000B00C0	32	R_HW1_IDX	r/w	120
0x000B00E0	32	R_HW2_IDX	r/w	120
0x000B0100	32	R_HW1_TX_CUR	r/w	105
0x000B0120	32	R_HW1_RX_CUR	r/w	105
0x000B0140	32	R_HW2_TX_CUR	r/w	106
0x000B0160	32	R_HW2_RX_CUR	r/w	106
0x000B0180	32	R_HW3_TX_CUR	r/w	106
0x000B01A0	32	R_HW3_RX_CUR	r/w	107
0x000B01C0	32	R_HW3_IDX	r/w	120
0x000B01E0	32	R_CODEC_IDX	r/w	121
0x000B01E4	16	R_ST_IDX	r/w	121
0x000B01E8	32	R_HW1_TS_EN	r/w	107
0x000B01EC	32	R_HW2_TS_EN	r/w	107
0x000B01F0	32	R_HW3_TS_EN	r/w	108
0x000B01F4	16	R_PFS0_CFG	r/w	108
0x000B01F6	16	R_PFS1_CFG	r/w	108
0x000B01F8	16	R_PFS2_CFG	r/w	109
0x000B01FA	16	R_PFS3_CFG	r/w	109
0x000B0200	8	R_PCM_CFG	r/w	110
0x000B0204	16	R_HW1_CTRL	r/w	111
0x000B0208	16	R_HW2_CTRL	r/w	113
0x000B020C	16	R_HW3_CTRL	r/w	115
0x000B0210	32	R_HW_SL_CNT	r	116
0x000C0000	32	R_ST_CFG	r/w	72
0x000C0004	32	R_ST_TX_STATUS	r	74
0x000C0008	32	R_ST_RX_STATUS	r	74
0x000C000C	32	R_ST_B12_IRQ_STATUS	r/w	77
0x000C0010	16	R_ST_D_FIFO_STATUS	r/w	78
0x000C0014	32	R_ST_B12_IRQ_EN	r/w	78
0x000C0018	16	R_ST_D_IRQ_EN	r/w	78
0x000C001C	32	R_ST_B1_TX_FIFO	r/w	79
0x000C0020	32	R_ST_B2_TX_FIFO	r/w	79
0x000C0024	32	R_ST_D_TX_FIFO	r/w	80
0x000C0028	32	R_ST_B1_RX_FIFO	r/w	80
0x000C0030	32	R_ST_B2_RX_FIFO	r/w	81
0x000C0034	32	R_ST_D_RX_FIFO	r/w	81
0x000C0038	16	R_ST_B1_CRC	r/w	81
0x000C003C	16	R_ST_B2_CRC	r/w	82
0x000C0040	16	R_ST_D_CRC	r/w	82
0x000C0044	8	R_ST_CTRL	r/w	83
0x000C0048	32	R_ST_B12_STATUS	r	84
0x000C004C	16	R_ST_D_STATUS	r	84
0x000C00C0	8	R_ST_RD_STATES	r	85
0x000C00C0	8	R_ST_WR_STATES	w	86
0x000C00C4	8	R_ST_CTRL1	w	87
0x000C00C8	8	R_ST_CTRL2	w	88
0x000C00CC	8	R_ST_CTRL3	w	88
0x000C00D0	8	R_ST_SQ_MF	r/w, r, w	89
0x000C00DC	8	R_ST_CLK_CTRL	w	89
0x000C00F0	8	R_ST_B1_RX	r	90
0x000C00F4	8	R_ST_B1_TX	w	90

Address	Width	Name	Mode	Page
0x000C00F8	8	R_ST_B2_RX	r	90
0x000C00FC	8	R_ST_B2_TX	w	90
0x000C0100	8	R_ST_D_RX	r	91
0x000C0104	8	R_ST_D_TX	w	91
0x000C0108	8	R_ST_E_RX	r	91
0x000D0000	32	R_CODEC_TX	r/w	123
0x000D0004	32	R_CODEC_RX	r	123
0x000D0008	32	R_CODEC_RX8	r	124
0x000D000C	32	R_CODEC_CTRL	r/w	125
0x000E0000	8	R_USB_ADDR	r/w	147
0x000E0004	8	R_USB_CFG	r/w, r	148
0x000E0008	8	R_USB_CTRL	r/w	148
0x000E000C	8	R_USB_EV1	r	149
0x000E0010	8	R_USB_EVMSK1	r/w	150
0x000E0014	8	R_USB_EV2	r	150
0x000E0018	8	R_USB_EVMSK2	r/w	151
0x000E0020	8	R_USB_OEP_SEL	r/w	151
0x000E0024	8	R_USB_ODATA	r	151
0x000E0028	8	R_USB_OCMD	r/w	152
0x000E002C	8	R_USB_OSTATUS	r	152
0x000E0030	8	R_USB_IEP_SEL	r/w	153
0x000E0034	8	R_USB_IDATA	w	153
0x000E0038	8	R_USB_ICMD	r/w	154
0x000E003C	8	R_USB_ISTATUS	r	154
0x000E0040	8	R_USB_OEP_EN	r/w, r	155
0x000E0044	8	R_USB_IEP_EN	r/w, r	155
0x000E0048	8	R_USB_OEP_STALL	r/w	156
0x000E004C	8	R_USB_IEP_STALL	r/w	156
0x000E0050	8	R_USB_OEP_EV	r	157
0x000E0054	32	R_USB_OEP_EVMSK	r/w	157
0x000E0058	32	R_USB_IEP_EV	r/w	157
0x000E005C	32	R_USB_IEP_EVMSK	r/w	158

1 General description

1.1 System overview

The HFC-S active is a single-chip solution for ISDN telecommunication applications. The device is designed for the following applications:

- ISDN telephones with / without data port
- ISDN PABX and ISDN POTS terminal adapters
- ISDN USB terminal adapters
- ISDN RS 232 terminal adapters
- various other ISDN applications by using external peripherals (e.g. ISDN LAN router)

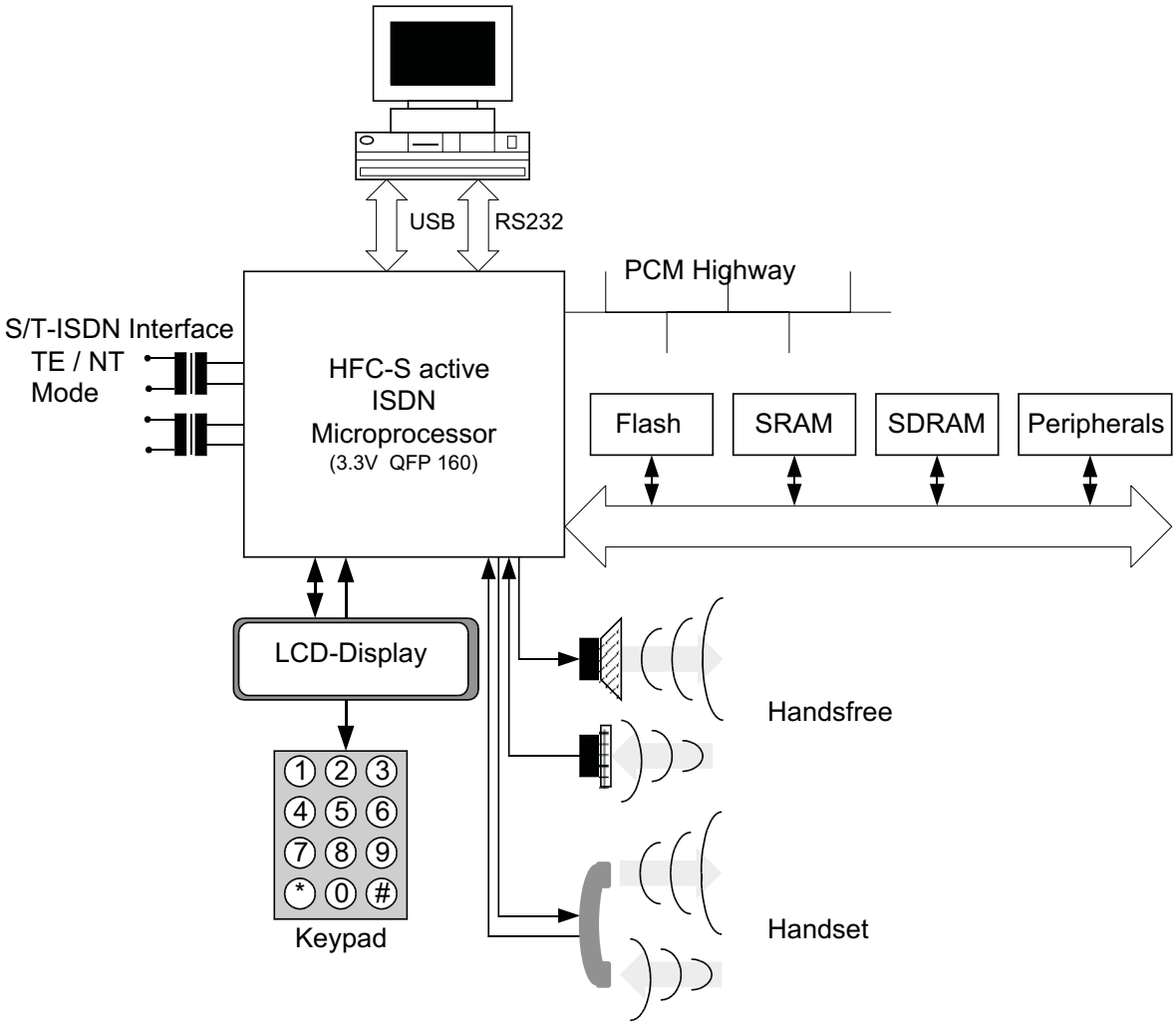


Figure 1: HFC-S active *application overview*

1.2 Features

- The HFC-S active contains a powerful 32 bit ARM7^{TM} RISC controller with a 32 bit address space operating at up to 61.440 MHz under worst case commercial conditions.
- Internal 16 kbyte SRAM (zero wait states)
- Supports 8 / 16 bit SRAM / Flash and 16 bit external SDRAM memory
- Advanced SDRAM controller with minimum wait states (full column burst mode)
- 5 independent external address spaces with software programmable wait state generation

- Full I.430 ITU S/T ISDN support in TE and NT mode
- Integrated ISDN S/T-controller with B- and D-channel HDLC support
- 6 independent read and write HDLC-controllers for B1-, B2- and D-channel
- B1-, B2- and D-channel transparent mode independently selectable
- Integrated FIFOs with 64 byte per channel and direction

- 2 integrated audio CODECs for the connection of analog devices (e.g. phone, fax, answering machine in PABX applications or handset in telephone applications)

- 3 independently programmable PCM highways with programmable switching unit between the 3 x 32 PCM highway channels (time slots), the B1- and B2-channels of the S/T-interface and the 2 CODECs
- Integrated high speed RS 232 interface (UART) with programmable data rate from 1.2 kbaud to 230.4 kbaud (theoretical maximum data rate: 1/8 of the system frequency f_{sys})
- ROM code for UART boot option integrated
- Full speed 12 Mbps Universal Serial Bus (USB) interface integrated compliant to USB specification 1.1 with
  - 4 input data endpoints with 64 byte FIFO each
  - 4 output data endpoints with 64 byte FIFO each
  - Bidirectional control endpoint with 16 byte FIFO per direction
  - On-chip USB transceiver
  - Control, interrupt and bulk transfer types
  - Bidirectional half-duplex link
  - Serial bus interface engine with packet decoding / generation, CRC generation and checking, NRZI encoding / decoding and bit-stuffing
- 2 programmable 16 bit timers with 8 bit prescaler with interrupt capability
- 10 bit programmable pulse width modulator (PWM) with interrupt capability
- Watchdog timer with interrupt capability and reset generation capability
- Up to 31 GPIO pins, 16 of these with interrupt capability
- Flexible and efficient interrupt processing for all system modules

- 3.3 V CMOS technology
- operation temperature range 0 °C ... +70 °C
- PQFP 160 case

1.3 Functional description

The HFC-S active is an ISDN telecommunication microprocessor system on a single chip (SoC). It is based on a powerful 32 bit ARM7^{TM} RISC processor with 16 bit and 32 bit instruction set. This industrial standard processor includes on-chip debugging facilities (embedded ICE).

The HFC-S active includes a 16 kbyte high speed memory, a S/T interface with layer 1 and layer 2 functions for the D-, B1- and B2-channel, a full speed USB-interface and a standard RS 232 interface. The CPU can boot from external Flash or from the RS 232 interface. The following block diagram illustrates the powerful architecture of the HFC-S active.

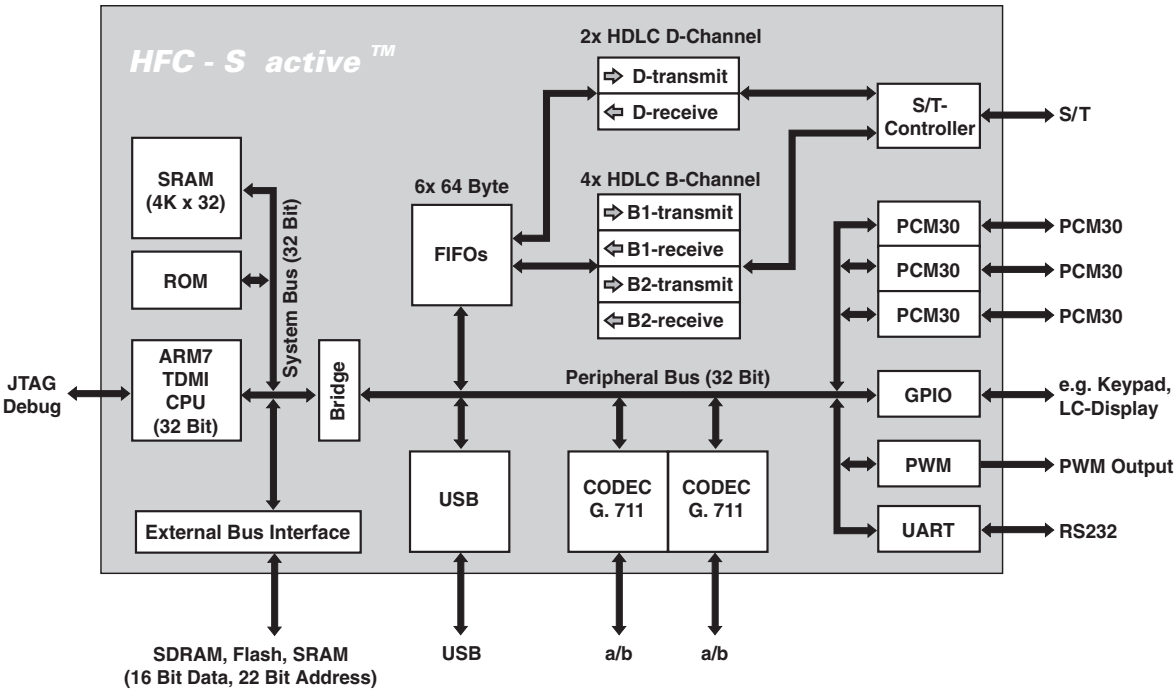


Figure 2: HFC-S active *block diagram*

The integrated CODECs allow the connection to telephone hand-sets or POTS ports e.g. for PABX applications. The CODECs have a programmable power-down mode. A processor controlled power management is supported. The programmable PLL allows to vary the system clock speed in the range from 12.288 MHz to 61.440 MHz.

1.4 Address space

The HFC-S active has an internal 4 k x 32 SRAM (16 kbyte) with the address range from 0x00000 to 0x3FFF. The address area 0x00000 to 0x0001F is reserved for exception handlers like shown in figure 3. The remaining SRAM can be used by application programs.

The internal ROM is divided into two sections:

- An 8 kbyte ROM which contains the first level boot loader and
- the 512 kbyte I/O area where the HFC-S active registers are located.

External SRAM, FLASH, SDRAM and peripherals are located at the address areas shown in figure 3. All other addresses are not used with the HFC-S active.

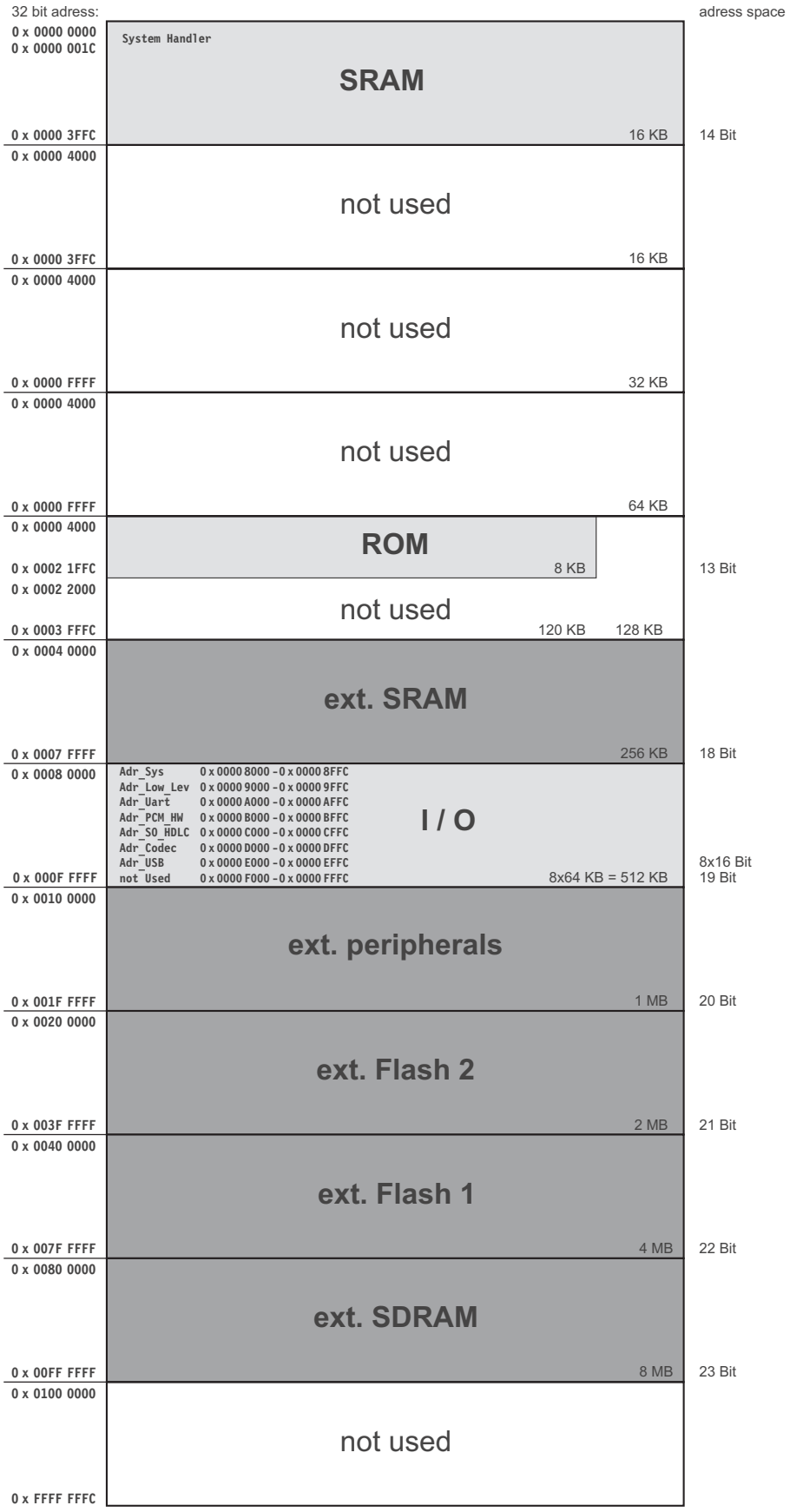


Figure 3: *Address space of* HFC-S active

1.5 Pin description

Pins with primary / secondary function are marked in figure 4. Table 1 shows an overview of these pins. The initial value of the register bits select always the primary function. The bit value has to be toggled to switch to the secondary function of the corresponding pin.

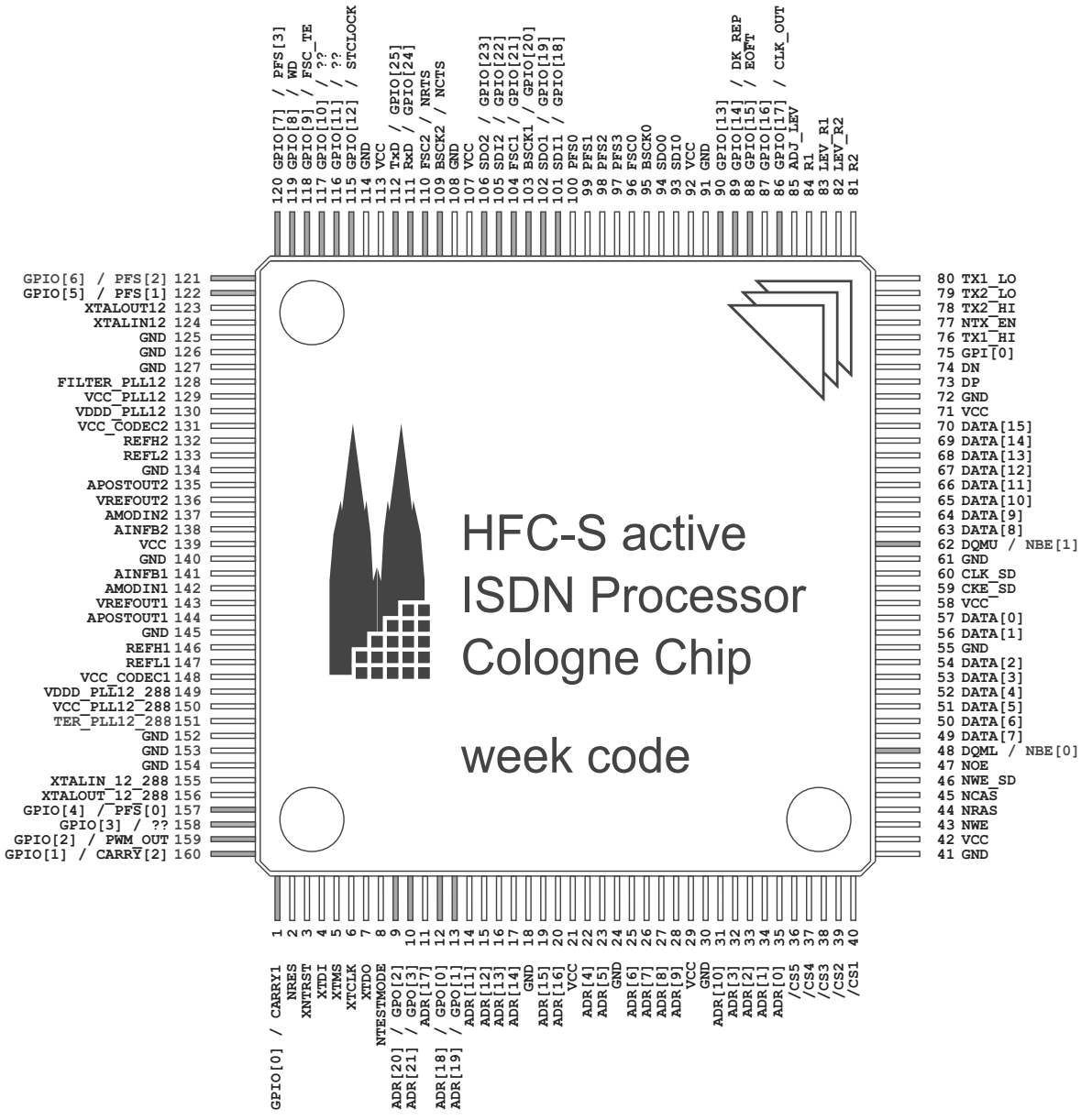


Figure 4: HFC-S active *pinout*

Table 1: *Overview of primary / secondary function pins and committed registers*

Pin number	Primary function	Secondary function	Register name	Bit name
1	CARRY1	GPIO0	R_GPIO_CTRL1	V_GPIO0_TI1
9	A20	GPO2	R_GPO_CTRL	V_GPO2_EN
10	A21	GPO3	R_GPO_CTRL	V_GPO3_EN
12	A18	GPO0	R_GPO_CTRL	V_GPO0_EN
13	A19	GPO1	R_GPO_CTRL	V_GPO1_EN
86	CLK_OUT	GPIO17	R_GPIO_CTRL2	V_GPIO17_CNT1B
88	EOFT	GPIO15	R_GPIO_CTRL1	V_GPIO15_EOFT
89	DK_REP	GPIO14	R_GPIO_CTRL1	V_GPIO14_DKREP
90	DK_EN	GPIO13	R_GPIO_CTRL1	V_GPIO13_DKEN
101	SDI1	GPIO18	R_GPIO_CTRL2	V_GPIO18_EN
102	SDO1	GPIO19	R_GPIO_CTRL2	V_GPIO19_EN
103	BCLK1	GPIO20	R_GPIO_CTRL2	V_GPIO20_EN
104	FSC1	GPIO21	R_GPIO_CTRL2	V_GPIO21_EN
105	SDI2	GPIO22	R_GPIO_CTRL2	V_GPIO22_EN
106	SDO2	GPIO23	R_GPIO_CTRL2	V_GPIO23_EN
109	BCLK2	/CTS	R_GPIO_CTRL2	V_NCTS_EN
110	FSC2	/RTS	R_GPIO_CTRL2	V_NRTS_EN
111	RXD	GPIO24	R_GPIO_CTRL2	V_GPIO24_EN
112	TXD	GPIO25	R_GPIO_CTRL2	V_GPIO25_EN
115	CLK_ST	GPIO12	R_GPIO_CTRL1	V_GPIO12_CNT1A
116	CLK_EXT	GPIO11	R_GPIO_CTRL1	V_GPIO11_CNT1B
117		GPIO10	R_GPIO_CTRL1	V_GPIO10_FSC_CONST
118	FSC_TE	GPIO9	R_GPIO_CTRL1	V_GPIO9_FSC_ST
119	WDT	GPIO8	R_GPIO_CTRL1	V_GPIO8_WD
120	PFS3	GPIO7	R_GPIO_CTRL1	V_GPIO7_PFS3
121	PFS2	GPIO6	R_GPIO_CTRL1	V_GPIO6_PFS2
122	PSF1	GPIO5	R_GPIO_CTRL1	V_GPIO5_PFS1
157	PFS0	GPIO4	R_GPIO_CTRL1	V_GPIO4_PFS0
158		GPIO3	R_GPIO_CTRL1	V_GPIO3_FSC
159	PWM_OUT	GPIO2	R_GPIO_CTRL1	V_GPIO2_PWM
160	CARRY2	GPIO1	R_GPIO_CTRL1	V_GPIO1_TI2

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
1	1st function 2nd function	CARRY1 GPIO0	O I / O	timer 1 carry signal general purpose input / output	LVCMOS	4 (SL)
2		/RES	I	chip reset (active low)	LVCMOS	
3		/XTRST	I	test reset (active low)	LVCMOS (PU)	
4		XTDI	I	test data input	LVCMOS (PU)	
5		XTMS	I	test mode select	LVCMOS (PU)	
6		XTCLK	I	test clock	LVCMOS (PU)	
7		XTDO	O	test data output		2 (tri)
8		DVCC		digital power supply		
9	1st function 2nd function	A20 GPO2	O O	external address bus general purpose output		8 8
10	1st function 2nd function	A21 GPO3	O O	external address bus general purpose output		8 8
11		A17	O	external address bus		8
12	1st function 2nd function	A18 GPO0	O O	external address bus general purpose output		8 8
13	1st function 2nd function	A19 GPO1	O O	external address bus general purpose output		8 8
14		A11	O	external address bus		8
15		A12	O	external address bus		8
16		A13	O	external address bus		8
17		A14	O	external address bus		8
18		DGND		digital ground		
19		A15	O	external address bus		8
20		A16	O	external address bus		8
21		DVCC		digital power supply		
22		A4	O	external address bus		8
23		A5	O	external address bus		8
24		DGND		digital ground		
25		A6	O	external address bus		8
26		A7	O	external address bus		8
27		A8	O	external address bus		8

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
28		A9	O	external address bus		8
29		DVCC		digital power supply		
30		DGND		digital ground		
31		A10	O	external address bus		8
32		A3	O	external address bus		8
33		A2	O	external address bus		8
34		A1	O	external address bus		8
35		A0	O	external address bus		8
36		/CS5	O	chip select for external SDRAM (active low)		8
37		/CS4	O	chip select for external peripherals (active low)		8
38		/CS3	O	chip select for external Flash 2 (active low)		8
39		/CS2	O	chip select for external Flash 1 (active low)		8
40		/CS1	O	chip select for external SRAM (active low)		8
41		DGND		digital ground		
42		DVCC		digital power supply		
43		/WE	O	write strobe for external asynchronous memories (active low)		O
44		/RAS	O	row address strobe for the external SDRAM (active low)		8
45		/CAS	O	column address strobe for the external SDRAM (active low)		8
46		/WE_SD	O	write strobe for the SDRAM (active low)		8
47		/OE	O	output enable for external asynchronous memories (active low)		8
48		DQML	O	low byte write mask for SDRAM		8
49		D7	I / O	external data bus	LVCMOS	8
50		D6	I / O	external data bus	LVCMOS	8
51		D5	I / O	external data bus	LVCMOS	8

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
52		D4	I / O	external data bus	LVCMOS	8
53		D3	I / O	external data bus	LVCMOS	8
54		D2	I / O	external data bus	LVCMOS	8
55		DGND		digital ground		
56		D1	I / O	external data bus	LVCMOS	8
57		D0	I / O	external data bus	LVCMOS	8
58		DVCC		digital power supply		
59		CKE_SD	O	clock enable for SDRAM		8
60		CLK_SD	O	clock for SDRAM		8
61		DGND		digital ground		
62		DQMU	O	high byte write mask for SDRAM		8
63		D8	I / O	external data bus	LVCMOS	8
64		D9	I / O	external data bus	LVCMOS	8
65		D10	I / O	external data bus	LVCMOS	8
66		D11	I / O	external data bus	LVCMOS	8
67		D12	I / O	external data bus	LVCMOS	8
68		D13	I / O	external data bus	LVCMOS	8
69		D14	I / O	external data bus	LVCMOS	8
70		D15	I / O	external data bus	LVCMOS	8
71		DVCC		digital power supply		
72		DGND		digital ground		
73		USB+	I / O	differential USB port (positive)	USB	USB
74		USB-	I / O	differential USB port (negative)	USB	USB
75		GPI0	I	general purpose input pin	LVCMOS (PD)	
76		TX1_HI	O	transmit port (high) for the S/T interface		S/T
77		/TX_EN	O	transmit enable port		S/T
78		TX2_HI	O	transmit port (high) for the S/T interface		S/T
79		TX2_LO	O	transmit port (low) for the S/T interface		S/T

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
80		TX1_LO	O	transmit port (low) for the S/T interface		S/T
81		R2	I	receive port for the S/T interface	S/T	
82		LEV_R2	I	level detect for R2	S/T	
83		LEV_R1	I	Level detect for R1	S/T	
84		R1	I	receive port for the S/T interface	S/T	
85		ADJ_LEV		adjust level control for the S/T interface		S/T
86	1st function	CLK_OUT	O	system clock f_{sys}		
	2nd function	GPIO17	I / O	general purpose input / output pin	LVCMOS	4 (SL)
87		GPIO16	I / O	general purpose input / output pin	LVCMOS	4 (SL)
88	1st function	EOFT	O	EOFT signal of the S/T interface		
	2nd function	GPIO15	I / O	general purpose input / output pin	LVCMOS	4 (SL)
89	1st function	DK_REP	O	DK_REP signal of the S/T interface		
	2nd function	GPIO14	I / O	general purpose input / output pin	LVCMOS	4 (SL)
90	1st function	DK_EN	O	DK_EN signal of the S/T interface		
	2nd function	GPIO13	I / O	general purpose input / output pin	LVCMOS	4 (SL)
91		DGND		digital ground		
92		DVCC		digital power supply		
93		SDI0	I	serial data input for PCM highway 1	LVCMOS	
94		SDO0	O	serial data output for PCM highway 1		2 (tri)
95		BCLK0	O	bit clock for PCM highway 1		2 (tri)
96		FSC0	I / O	frame sync signal for PCM highway 1	LVCMOS	2 (tri)
97		PFS3	O	peripheral frame sync signal		2
98		PFS2	O	peripheral frame sync signal		2

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
99		PFS1	O	peripheral frame sync signal		2
100		PFS0	O	peripheral frame sync signal		2
101	1st function	SDI1	I	serial data input for PCM highway 2	LVCMOS	2
	2nd function	GPIO18	I / O	general purpose input / output	LVCMOS	2
102	1st function	SDO1	O	serial data output for PCM highway 2	LVCMOS	2 (tri)
	2nd function	GPIO19	I / O	general purpose input / output	LVCMOS	2
103	1st function	BCLK1	O	bit clock for PCM highway 2	LVCMOS	2 (tri)
	2nd function	GPIO20	I / O	general purpose input / output	LVCMOS	2
104	1st function	FSC1	I / O	frame sync signal for PCM highway 2	LVCMOS	2 (tri)
	2nd function	GPIO21	I / O	general purpose input / output	LVCMOS	2
105	1st function	SDI2	I	serial data input for PCM highway 3	LVCMOS	2
	2nd function	GPIO22	I / O	general purpose input / output	LVCMOS	2
106	1st function	SDO2	O	serial data output for PCM highway 3	LVCMOS	2 (tri)
	2nd function	GPIO23	I / O	general purpose input / output	LVCMOS	2
107		DVCC		digital power supply		
108		DGND		digital ground		
109	1st function	BCLK2	O	bit clock for PCM highway 2	LVCMOS	2 (tri)
	2nd function	/CTS	I	CTS signal (UART)		
110	1st function	FSC2	I / O	frame sync signal for PCM highway 3		2 (tri)
	2nd function	/RTS	O	RTS signal (UART)		
111	1st function	RXD	I	serial receive data (UART)	LVCMOS	2
	2nd function	GPIO24	I / O	general purpose input / output		
112	1st function	TXD	O	serial transmit data (UART)	LVCMOS	2
	2nd function	GPIO25	I / O	general purpose input / output		
113		DVCC		digital power supply		

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
114		DGND		digital ground		
115	1st function	CLK_ST	O	S/T clock f_{ISDN}		
	2nd function	GPIO12	I / O	general purpose input / output	LVCMOS	4 (SL)
116	1st function	CLK_EXT	O	clock for external devices (f_{ext})		
	2nd function	GPIO11	I / O	general purpose input / output	LVCMOS	4 (SL)
117	1st function		O			
	2nd function	GPIO10	I / O	general purpose input / output	LVCMOS	4 (SL)
118	1st function	FSC_TE	O	FSC_TE signal of the S/T interface		
	2nd function	GPIO9	I / O	general purpose input / output	LVCMOS	4 (SL)
119	1st function	WDT	O	carry signal of the watchdog timer		
	2nd function	GPIO8	I / O	general purpose input / output	LVCMOS	4 (SL)
120	1st function	PFS3	O	peripheral frame sync 3 signal with interrupt capability		
	2nd function	GPIO7	I / O	general purpose input / output	LVCMOS	4 (SL)
121	1st function	PFS2	O	peripheral frame sync 2 signal with interrupt capability		
	2nd function	GPIO6	I / O	general purpose input / output	LVCMOS	4 (SL)
122	1st function	PSF1	O	peripheral frame sync 1 signal with interrupt capability		
	2nd function	GPIO5	I / O	general purpose input / output	LVCMOS	4 (SL)
123		XTALOUT2	O	Output for the USB quartz oscillator		XTAL
124		XTALIN2	I	Input for the USB quartz oscillator	XTAL	
125		DGND		digital ground		
126		AGND		analog / digital sub-bias power		
127		AGND		analog ground		
128		C_PLL2		PLL filter capacitance	analog	analog
129		AVCC_PLL2		analog power supply		

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
130		DVCC_PLL2		digital power supply		
131		AVCC_CODEC2		analog power supply		
132		REFH1			analog	
133		REFL1			analog	
134		AGND		analog ground (0.0V)		
135		APOSTOUT1				analog
136		VREFOUT1				analog
137		AMODIN1			analog	
138		AINFB1				analog
139		DVCC_CODEC		digital power supply		
140		DGND		digital ground		
141		AINFB0				analog
142		AMODIN0			analog	
143		VREFOUT0				analog
144		APOSTOUT0				analog
145		AGND		analog ground (0.0V)		
146		REFH0				analog
147		REFL0			analog	
148		AVCC_CODEC1		analog power (+3.3V)		
149		DVCC_PLL1		digital power supply		
150		AVCC_PLL1		analog power supply		
151		C_PLL1			analog	analog
152		AGND		analog ground		
153		AGND		analog / digital sub-bias power		
154		DGND		digital ground		
155		XTALIN1			XTAL	
156		XTALOUT1				XTAL
157	1st function	PFS0	O	peripheral frame sync 0 signal with interrupt capability		
	2nd function	GPIO4	I / O	general purpose input / output	LVCMOS	4 (SL)

(continued on next page)

(continued from previous page)

Pin	Function	Name	I/O	Description	U_{in} / V	I_{out} / mA
158	1st function		O			
	2nd function	GPIO3	I / O	general purpose input / output	LVCMOS	4 (SL)
159	1st function	PWM_OUT	O	PWM output	LVCMOS	4 (SL)
	2nd function	GPIO2	I / O	general purpose input / output	LVCMOS	4 (SL)
160	1st function	CARRY2	O	Timer 2 carry signal		
	2nd function	GPIO1	I / O	general purpose input / output	LVCMOS	4 (SL)

Legend:

SL: slew rate controlled output pad

OD: open drain output pad

tri: tristate output pad

PU: pullup resistor integrated in input pad

2 CPU, memory and bus interface

2.1 ARM7^{TM} CPU

The HFC-S active is based on the ARM7 TDMI processor core revision 1b[1], which is a member of the Advanced RISC Machines (ARM) family of general purpose 32 bit microprocessors. The ARM7 TDMI offers high performance at a very low power consumption and cost.

The ARM7^{TM} architecture is based on Reduced Instruction Set Computer (RISC) principles. The instruction set and related decode mechanisms are much simpler than those of microprogrammed Complex Instruction Set Computers (CISC). This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective chip.

Pipelining is employed so that all parts of the processing and memory system can operate continuously. Typically, while one instruction is being executed, its successor is being decoded and a third instruction is being fetched from memory.

The ARM7^{TM} memory interface has been designed to allow the performance potential to be realized without incurring high costs in the memory system. Speed-critical control signals are pipelined to allow system control functions to be implemented in standard low-power logic. These control signals facilitate the exploitation of the fast local access modes offered by industry standard dynamic RAMs.

The HFC-S actives ARM7^{TM} CPU including its bus interface is fixed to little endian mode.

Misaligned memory access should always be avoided. The internal memory and external asynchronous memory round down to the next aligned address. The SDRAM controller replaces an unaligned access by several memory accesses, but at page boundaries there might be unexpected results.

The access to not used memory areas returns undefined values.

More about the ARM7 TDMI CPU can be looked up on the following WWW sites:

`http://www.arm.com/Documentation/UserMans/PDF/ARM7TDMI.html`

2.2 External bus interface

2.2.1 Overview

The HFC-S active contains a versatile interface for external memories. Up to four external memory components can be connected at the same time and one additional address region is reserved for external peripheral chips or I/O expansions. The external memory bus is 16 bit wide and allows to connect 8 bit and 16 bit memory components or external peripheral devices, except the SDRAM which has always 16 bit bus width. The software can set the number of waitstates and the bus width (8 / 16 bit) for each address area individually. Five chip select signals are generated for the following purposes:

- 1 SDRAM of up to 4 M x 16 (8 Mbyte / 64 Mbit)
- 1 Flash of up to 4 M x 8 or 2 M x 16 (4 Mbyte / 32 Mbit)
- 1 Flash of up to 2 M x 8 or 1 M x 16 (2 Mbyte / 16 Mbit)

[1] ARM7 TDMI is a registered trademark of ARM Ltd.

Table 3: *Overview of the* HFC-S active *external bus interface pins (* : Primary function)*

Number	Name	Description
35	A0	external address bus
34	A1	
33	A2	
32	A3	
22	A4	
23	A5	
25	A6	
26	A7	
27	A8	
28	A9	
31	A10	
14	A11	
15	A12	
16	A13	
17	A14	
19	A15	
20	A16	
11	A17	
12	A18	
13	A19 *	
9	A20	
10	A21 *	
40	/CS1	chip select for external SRAM (active low)
39	/CS2	chip select for external Flash 1 (active low)
38	/CS3	chip select for external Flash 2 (active low)
37	/CS4	chip select for external peripherals (active low)
36	/CS5	chip select for external SDRAM (active low)

Number	Name	Description
57	D0	external data bus
56	D1	
54	D2	
53	D3	
52	D4	
51	D5	
50	D6	
49	D7	
63	D8	
64	D9	
65	D10	
66	D11	
67	D12	
68	D13	
69	D14	
70	D15	
47	/OE	output enable for external asynchronous memories (active low)
43	/WE	write strobe for external asynchronous memories (active low)
60	CLK_SD	clock for SDRAM
59	CKE_SD	clock enable for SDRAM
46	/WE_SD	write strobe for the SDRAM (active low)
45	/CAS	column address strobe for the external SDRAM (active low)
44	/RAS	row address strobe for the external SDRAM (active low)
62	DQMU	high byte write mask for SDRAM
48	DQML	low byte write mask for SDRAM

- 1 SRAM of up to 256 k x 8 or 128 k x 16 (256 kbyte / 2 Mbit)
- 1 area for external peripherals of up to 1 M x 8 or 512 k x 16 (1 Mbyte / 8 Mbit)

Table 3 shows all pins of the HFC-S active which are in the context of chapter 2.2.

For applications requiring more external SRAM memory, the areas for SRAM and external peripherals can be swapped by swapping the chip select signals (allowing 1 Mbyte of SRAM).

Table 4: *Overview of the* HFC-S active *external bus interface registers*

Address	Name	Page
0x00080018	R_WS1	37
0x0008001C	R_WS2	38
0x00080010	R_SDRAM_CTRL	36

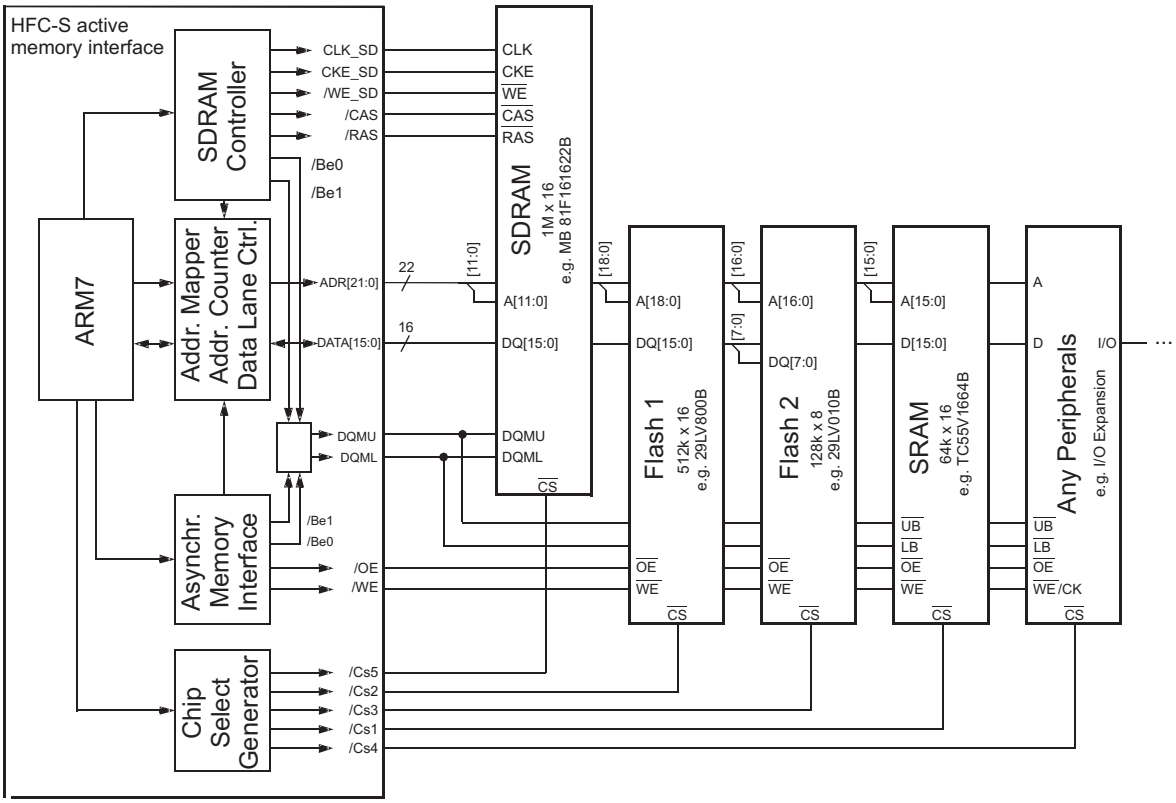


Figure 5: *Connecting external memory components to the* HFC-S active *(example)*

Flash memory, SRAM and external peripherals can be connected via the asynchronous memory interface which is described in section 2.2.2. The SDRAM needs additional control signals. A detailled description of the SDRAM controller can be found in section 2.2.3.

Figure 5 shows an example schematic with the connection of one SDRAM, one 16 bit Flash, one 8 bit Flash and one SRAM to the HFC-S active. Due to the used RAM sizes which are smaller than the maximum sizes, the shown address ranges are underutilized in this example. The SDRAM address is a 12 bit multiplexed signal for alternate row and column address.

2.2.2 Asynchronous memory interface (Flash, SRAM, external peripherals)

8 / 16 bit bus width options

The asynchronous memory interface generates /OE and /WE signals at the pins 47 and 43 that allow direct interfacing to external Flash, SRAM and peripheral devices.

For all external components except the SDRAM (which is always 16 bit wide), a bus width of 8 bit or 16 bit can be selected individually. 32 bit, 16 bit and 8 bit read and write access is fully supported on 16 bit memories as well as on 8 bit memories. If the access size is greater than the memory size, the internal logic of HFC-S active will perform 2 or 4 memory access cycles with automatic address incrementing and data lane selection.

External 16 bit memories should be connected to the external address bus bits A[n:0] as the internal ARM7^{TM} address bits A[n+1:1] are shifted down by one bit by the internal logic for each access to an external 16 bit memory. This allows an optimum usage of the address signals.

When writing an 8 bit data byte to a 16 bit memory, data qualifier signals[2] (DQMU, DQML at pins 62, 48) are generated to mask the 8 data bits that should not be written.

Waitstates programming

The number of waitstates can be selected individually for each external component, except the SDRAM (which always operates at the maximum frequency) in order to comply with the different timing of the memory components and peripherals. For a selected number of n waitstates, the access time for one memory access cycle will be $(n + 1)$ ARM7^{TM} clock periods.

If the ARM7^{TM} access size is greater than the memory size, the total access time will be 2 or 4 times the access time for a single memory access cycle because 2 or 4 memory accesses are performed. The total number of clock cycles used for a memory access can be determined as follows (for n waitstates selected):

- an 8 bit access on an 8 bit memory takes $n + 1$ clock cycles,
- a 16 bit access on an 8 bit memory takes $2(n + 1)$ clock cycles (2 memory accesses of 8 bit),
- a 32 bit access on an 8 bit memory takes $4(n + 1)$ clock cycles (4 memory accesses of 8 bit),
- an 8 bit access on a 16 bit memory takes $n + 1$ clock cycles,
- a 16 bit access on a 16 bit memory takes $n + 1$ clock cycles,
- a 32 bit access on a 16 bit memory takes $2(n + 1)$ clock cycles (2 memory accesses of 16 bit),
- a register access (32 bit) takes $n + 1$ clock cycles,
- a 32 bit access on the internal SRAM and ROM is always performed without wait states.

Some internal modules of the HFC-S active have also a wait states programming register. The internal SRAM, the S/T interface, The CODECs and the USB interface have indepent waitstates bitmaps in the register R_WS2, while all other modules have the same waitstates value. Larger waitstates values might be useful to reduce the CPU load.

External bus interface timing

The example in figure 6 shows a 16 bit read access on an external 8 bit Flash component with 2 waitstates, followed by a 16 bit write access to the same external 8 bit Flash component with 2 waitstates. Please note that the total access time is 6 clock cycles in this case (access 2 words of 8 bit; 2 waitstates = 3 clock cycles for each access). For a high performance system, a Flash with 16 bit data bus and 0 waitstates could be used (if available), allowing a single cycle access.

[2] DQM: Data I/O Mask, DQMU: Upper byte of Data I/O Mask, DQML: Lower byte of Data I/O Mask

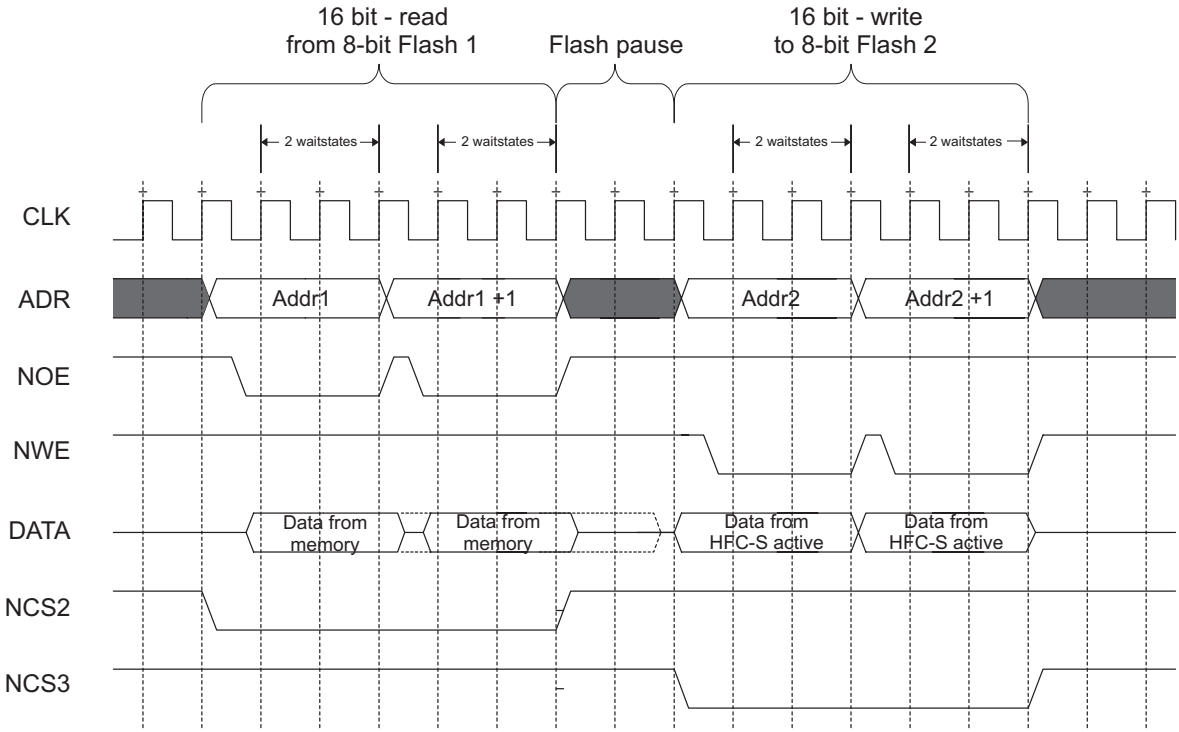


Figure 6: *External bus interface timing diagram (example with 2 programmed waitstates cycles)*

Avoiding bus contention (Flash pause)

As some Flash devices have a very long data bus release time (/OE high to output HIGH-Z), a pause (delay) can be programmed in order to avoid bus contention on the bidirectional external data bus (bitmap V_WS_FLASH_DL in register R_WS1). In figure 6, this pause is 2 clock cycles. It can be programmed in the range of 0 ... 3 cycles. The delay is only inserted between a read access cycle from the external Flash and a write access to any external memory and only when it is necessary to avoid bus contention. For example, if the two clock cycles following the Flash read are used for internal memory access, no delay is inserted. In the case of one internal cycle between a Flash read and a Flash write, only one wait cycle will be inserted in the case that a number of 2 cycles has been selected.

It is recommended to use the waitstates shown in table 5.

2.2.3 SDRAM controller

The HFC-S active has an advanced SDRAM controller which supports all SDRAMs with

- max. 8 MByte capacity,
- 16 bit data bus,
- and CAS [3] latency = 2.

The main feature of the SDRAM controller is the full column burst mode. The full column burst mode is controlled by the SEQ signal of the ARM7^{TM} CPU indicating sequential memory access

[3] CAS: Column Address Strobe

Table 5: *Recommended values for waitstates programming*

Register	Bitmap	Recommended Value
R_WS2 R_WS2 R_WS1	V_WS_ST V_WS_CODEC V_WS_PCM	not less than $\frac{f_{sys}}{12.288\,\mathrm{MHz}} - 1$
R_WS2	V_WS_USB	not less than $\frac{7 \cdot f_{sys}}{12.000\,\mathrm{MHz}} - 1$
R_WS2	V_WS_SRAM V_WS_GEN	should always be 0
R_WS1	V_WS_FLASH1 V_WS_FLASH2 V_WS_EXTIO V_WS_SRAM	depends on the access time of the external memory / peripheral devices

operations. This feature increases the software performance by 20% typically, if the program is running from the external SDRAM. The SDRAM controller is clock-controlled by the system clock (ARM7^{TM} clock).

After a system reset the SDRAM controller initialization sequence starts automatically. At the end of the initialization process the V_SDRAM_RDY bit of the register R_SDRAM_CTRL is automatically set to 1. The software has to take care that no access on the external bus is carried out during the SDRAM initialization.

The software can switch the SDRAM to a power down mode to reduce the power consumption of the ISDN application by setting V_SDRAM_EN in the R_SDRAM_CTRL register.

SDRAM interface timing

The SDRAM interface timing is shown in figure 7.

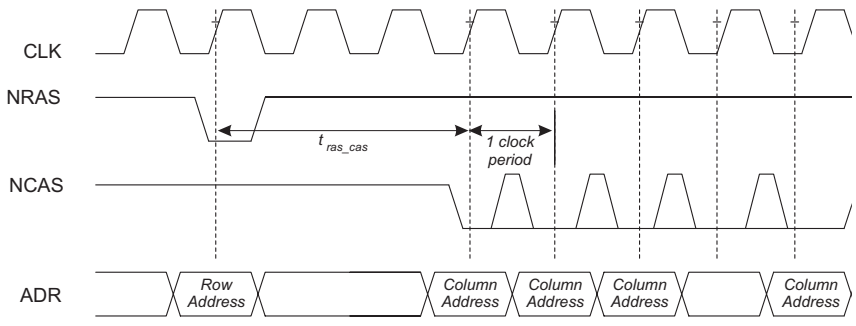


Figure 7: *SDRAM interface timing*

2.3 Boot loader

The *First Level Boot Loader* is a small programm at ROM address 0x20000 which will always be started after a system reset. It decides about the boot program address.

1. If there is the *magic number* 0x46352413 found at address 0x00040000 (FLASH bank 1), the boot program is expected from address 0x00040004 and will be startet at once. The magic number is not a valid instruction code.
2. If the magic number is not found, the HFC-S active waits for seriell data at the RS 232 port with a fixed baud rate of

$$f_{baud} = 57.600\,\mathrm{kbaud} \cdot \frac{f_{sys}}{12.288\ \mathrm{MHz}} .$$

   Note, that the system clock f_{sys} is always equal to the quartz frequency at this time. The received data is stored from address 0x00000400 (internal SRAM) and is started afterwards.
3. If there are no seriell data for one second, the boot loader terminates the wait loop and starts the program from address 0x00040000. Note, that in this case there is no magic number at the beginning of the FLASH bank 1.

2.4 Register description

R_SDRAM_CTRL	(read / write, read)	0x00080010
SDRAM control register		

Bits	Reset Value	Name	Description
11..0	0x27	**M_SDRAM_MO**	Shows the setting of the SDRAM mode register
12		**(reserved)**	Must be set to 0
13	0	**V_SDRAM_RDY**	Indicates the end of the SDRAM initialization '0' = SDRAM not ready '1' = SDRAM ready
14	0	**V_SDRAM_EN**	Activates the SDRAM controller '0' = power up '1' = power down
15	0	**V_BURST_EN**	Activates the full column burst mode '0' = disabled '1' = enabled

Note: Word accesses have to be word aligned on the page boundary.

R_WS1	(read / write)	0x00080018

1st waitstates register and bus width control for the external bus interface

Bits	Reset Value	Name	Description
3..0	2	**V_WS_FLASH1**	Sets the waitstates for the external Flash bank 1 (0 ... 15 clock cycles)
7..4	0xF	**V_WS_FLASH2**	Sets the waitstates for the external Flash bank 2 (0 ... 15 clock cycles)
11..8	0xF	**V_WS_EXTIO**	Sets the waitstates for the external peripherals (0 ... 15 clock cycles)
15..12		**(reserved)**	
19..16	0xF	**V_WS_SRAM**	Sets the waitstates for the external SRAM (0 ... 15 clock cycles)
24..20	0x0F	**V_WS_PCM**	Sets the waitstates for the PCM highway (3 ... 31 clock cycles)
25	0	**V_FLASH1_WORD**	Sets the data bus width for the external Flash bank 1 '0' = 8 bit '1' = 16 bit
26	0	**V_FLASH2_WORD**	Sets the data bus width for the external Flash bank 2 '0' = 8 bit '1' = 16 bit
27	1	**V_EXTIO_WORD**	Sets the data bus width for the external peripherals '0' = 8 bit '1' = 16 bit
28	0	**V_SRAM_WORD**	Sets the data bus width for the external SRAM '0' = 8 bit '1' = 16 bit
30..29	0	**V_WS_FLASH_DL**	Sets the delay between Flash read access and write access (0 ... 3 clock cycles)
31		**(reserved)**	

R_WS2	(read / write)	0x0008001C
2nd waitstates register		

Bits	Reset Value	Name	Description
4..0	0x02	**V_WS_ST**	Sets the waitstates for the S/T HDLC module (0 ... 31 clock cycles)
8..5	0	**V_WS_SRAM**	Sets the waitstates for the internal SRAM (0 ... 15 clock cycles)
11..9	0	**V_WS_GEN**	Sets the waitstates for all other interfaces (0 ... 7 clock cycles)
17..12	0x00	**V_WS_USB**	Sets the waitstates for the USB interface (0 ... 63 clock cycles)
21..18	0	**V_WS_CODEC**	Sets the waitstates for the CODEC (0 ... 15 clock cycles)
31..22		**(reserved)**	

3 Clocks, timer and interrupt

3.1 Clocks of the HFC-S active

Table 6: *Overview of the* HFC-S active *clock pins (all primary function)*

Number	Name	Description
86	CLK_OUT	system clock f_{sys}
115	CLK_ST	S/T clock f_{ISDN}
116	CLK_EXT	clock for external devices (f_{ext})

Table 7: *Overview of the* HFC-S active *clock registers*

Address	Name	Page	Address	Name	Page
0x00080020	R_PLL1_CFG	44	0x00080038	R_OSC_CFG	45
0x00080024	R_PLL2_CFG	44	0x00080040	R_CNT1B_CFG	46
0x00080028	R_DIV1_CFG	45			

3.1.1 Clock distribution

The HFC-S active contains a versatile clock distribution circuit including two crystal oscillators, two PLLs, a clock divider and two modulo counters for the generation of several, individually programmable clock frequencies from one or two external crystals. Section 3.1.4 shows a circuitry example which needs only one external crystal to put all subsystems into operation.

Figure 8 illustrates, how the functional blocks of the HFC-S active clock distribution work together. As the crystal frequencies are not prescribed, frequency values are only pointed out where they are fixed by specification requirements.

Table 8: *Suitable values for CNT 1A programming*

System clock frequency f_{sys}	CNT 1A M	CNT 1A N
12.288 MHz	1	1
24.576 MHz	1	2
36.864 MHz	1	3
49.152 MHz	1	4
61.440 MHz	1	5

Table 9: *Suitable values for PLL 2 programming*

Crystal frequency	PLL 2 P	PLL 2 M	PLL 2 S
8.000 MHz	4	64	1
12.000 MHz	7	64	1
12.288 MHz	14	117	1
16.000 MHz	10	64	1
24.576 MHz	30	117	1

Programmable system clock f_{sys}: The crystal which is connected to the OSC 1 block is used as a reference for the programmable frequency of the ARM7^{TM} CPU and some other subsystems like shown in figure 8. By using the PLL 1 and the divider DIV 1, the input frequency can be

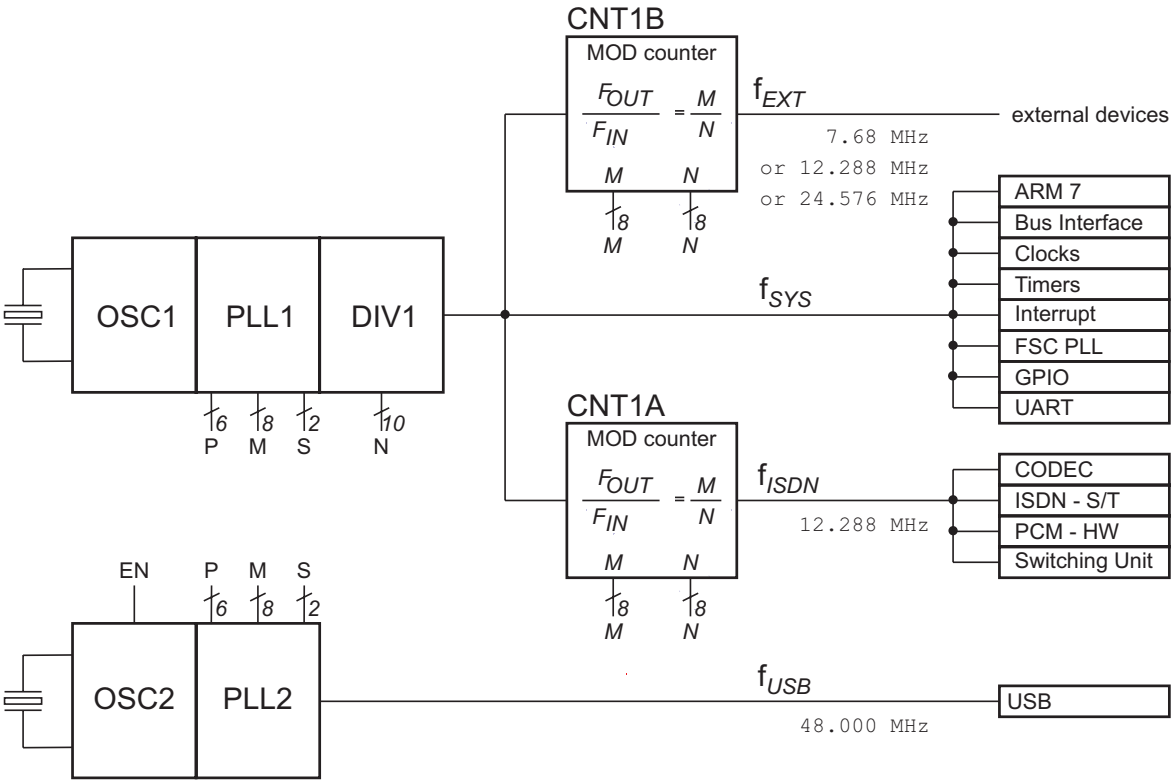


Figure 8: *Clock distribution in the* HFC-S active

Table 10: *Suitable values for CNT 1B programming*

System clock frequency f_{sys}	frequency f_{ext}	CNT 1B M	CNT 1B N
12.288 MHz	7.680 MHz	5	8
	12.288 MHz	1	1
	24.576 MHz	2	1
24.576 MHz	7.680 MHz	5	16
	12.288 MHz	1	2
	24.576 MHz	1	1
36.864 MHz	7.680 MHz	5	24
	12.288 MHz	1	3
	24.576 MHz	2	3
49.152 MHz	7.680 MHz	5	32
	12.288 MHz	1	4
	24.576 MHz	2	1
61.440 MHz	7.680 MHz	1	8
	12.288 MHz	1	5
	24.576 MHz	2	5

scaled up or down flexibly over a very wide frequency range from some kHz to 49.152 MHz. A detailed describtion of the f_{sys} generation is given in section 3.1.2.

12.288 MHz clock for ISDN related peripheral modules: Some peripheral modules of the HFC-S active like shown in figure 8 have to operate at a fixed frequency $f_{ISDN} = 12.288\,\mathrm{MHz}$ due to fixed data rate requirements. In order to allow interfacing to the ARM7^{TM} CPU, the clock skew between f_{ISDN} and the ARM7^{TM} clock f_{sys} has to be nearly zero. As the clock phase of the PLL output relative to the PLL input is uncontrollable, the original crystal frequency cannot be used to derive a clock signal for these modules.

These hardware limitations have been solved by implementing the programmable modulo counter CNT 1A. Some examples and the matching counter parameters are listed in table 8 [4].

The clock jitter of the modulo counter CNT 1A is max. $\pm 1/(2 \cdot f_{sys})$ in clock period. For an input frequency $f_{sys} = n \cdot 12.288\,\mathrm{MHz}$ with an integer n, the clock jitter is zero as the modulo counter behaves like a divider.

7.68 MHz / 12.288 MHz / 24.576 MHz clock for external S/T transceivers ICs: Besides the usage of additional, external Cologne Chip ISDN S/T controllers (such like HFC-S mini), the HFC-S active also supports the use of simple ISDN transceiver ICs with lower functional integration.

For the connection of such external S/T transceiver ICs to the HFC-S active, a synchronization frequency of $f_{ext} = 7.68\,\mathrm{MHz}$, 12.288 MHz or 24.576 MHz is required. The modulo counter CNT 1B has been implemented to generate this frequency. Tabel 10 shows some suitable programming parameters for several system frequencies. Due to the sophisticated dual slope design of this counter, the clock jitter of the output frequency is only max. $\pm 1/(4 \cdot f_{sys})$ in clock period. For an input frequency $f_{sys} = \frac{n}{2} \cdot f_{ext}$ with an integer n, the clock jitter is zero.

48.000 MHz clock for USB controller: The crystal which is connected to the OSC 2 block is only used to generate a frequency of $f_{USB} = 48.000\,\mathrm{MHz}$ needed for the USB controller. Crystals of different frequencies can be used to generate f_{USB} as the PLL is programmable. A detailed describtion of the f_{sys} generation is given in section 3.1.3.

3.1.2 Clock frequency selection and clock switching (system clock f_{sys})

The PLL block includes a crystal oscillator, the programmable PLL 1 and the programmable divider DIV 1. The PLL requieres an external capacitor of 820 pF. The frequency scaling ratio of the PLL and the divider can be selected by parameters as shown in figure 9. In addition, the PLL can be turned off (V_PLL1_PWRDN=1) to save energy.

The PLL and the divider can be selected or bridged by multiplexers individually. One multiplexer can select the PLL output (V_PLL1_SEL = 1) or the original oscillator frequency (V_PLL1_SEL = 0). As the clock phase of the PLL output relative to the PLL input is uncontrollable, the multiplexer is synchronized to the clock signals to avoid short clock pulses (spikes). For detailed timing information see appendix C.

For bridging the programmable divider after the PLL, a second multiplexer can select the divider output (V_DIV1_SEL = 1) or the PLL multiplexer output (V_DIV1_SEL = 0).

Suitable parameters for the PLL 1 block and DIV 1 block are given in table 11. Most PLL output frequencies can be achieved with various PLL parameter sets. It is recommended to select the shown PLL parameters to ensure a stable operation.

[4] Typically, for both counters CNT 1A and CNT 1B there are several parameter sets which result in the same output frequency. In these cases one should prefer small values for M and N.

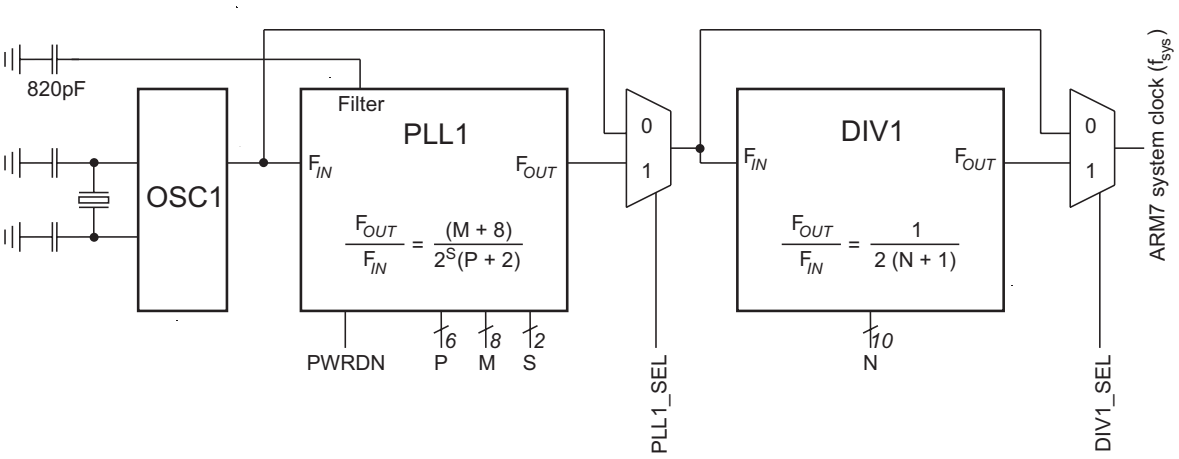


Figure 9: *Programmable PLL 1 block*

Table 11: *Suitable values for PLL 1 programming (DIV 1 disabled)*

Crystal frequency	desired system clock	PLL 1 P	PLL 1 M	PLL 1 S	Crystal frequency	desired system clock	PLL 1 P	PLL 1 M	PLL 1 S
	12.288 MHz	5	78	3		12.288 MHz	19	121	3
	24.576 MHz	5	78	2		24.576 MHz	19	121	2
8.000 MHz	36.864 MHz	7	75	1	16.000 MHz	36.864 MHz	21	98	1
	49.152 MHz	5	78	1		49.152 MHz	19	121	1
	61.440 MHz	9	161	1		61.440 MHz	23	184	1
	12.288 MHz	14	123	3		12.288 MHz	16	64	3
	24.576 MHz	14	123	2		24.576 MHz	16	64	2
12.000 MHz	36.864 MHz	12	78	1	24.576 MHz	36.864 MHz	16	46	1
	49.152 MHz	14	123	1		49.152 MHz	16	64	1
	61.440 MHz	15	166	1		61.440 MHz	16	82	1
	12.288 MHz	7	64	3					
	24.576 MHz	7	64	2					
12.288 MHz	36.864 MHz	7	46	1					
	49.152 MHz	7	64	1					
	61.440 MHz	7	82	1					

As the system clock f_{sys} is also the input signal for the modulo counters CNT 1A and CNT 1B, possible system clock frequencies are restricted by the requiered counter output frequencies.

3.1.3 Clock frequency selection and clock switching (USB clock)

The PLL 2 block includes a crystal oscillator and the programmable PLL 2 (see fig. 10). The PLL requieres an external capacitor of 820 pF. For details of the PLL and PLL multiplexer please refer to

the previous section. Some practicable crystal frequencies and the matching PLL 2 parameters are shown in table 9.

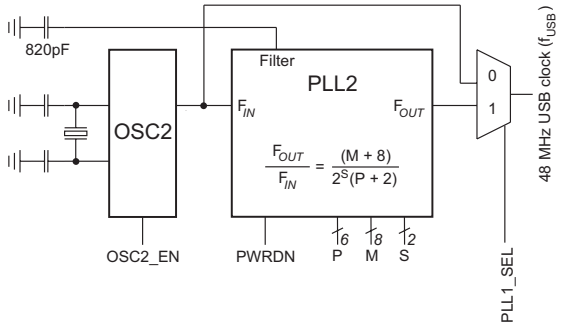


Figure 10: *Programmable PLL 2 block*

In addition to the PLL 2 block, the crystal oscillator OSC 2 can be turned off (V_OSC2_EN = 0) to save energy. A multiplexer allows to switch the OSC 2 signal directly to the f_{USB} output with the setting V_PLL2_SEL = 0.

3.1.4 USB clock generation from OSC 1

The USB clock can be derived from the OSC 1 crystal, if the second crystal connected to OSC 2 shall be saved. As there is no internal signal path between OSC 1 and OSC 2 resp. f_{sys} and f_{USB}, the OSC 1 output (pin XTALOUT1) must be connected to OSC 2 input (pin XTALIN2). Some exmaples of practicable PLL 2 parameters are given in Table 9.

3.1.5 Register description

R_PLL1_CFG	(read / write)	0x00080020

Configuration register for the PLL 1 (system clock)

Bits	Reset Value	Name	Description
7..0	0x00	**V_PLL1_M**	multiplier M for PLL 1 (range 0 … 255)
13..8	0x00	**V_PLL1_P**	divider P for PLL 1 (range 0 … 63)
15..14	0	**V_PLL1_S**	divider S for PLL 1 (range 0 … 3)
25..16	0x000	**V_DIV1_N**	divider N for DIV 1 (range 0 … 1023)
26	1	**V_PLL1_PWRDN**	set the PLL 1 in power down mode '0' = power on '1' = power down
31..27		**(reserved)**	

R_PLL2_CFG	(read / write)	0x00080024

Configuration register for the PLL 2 (USB clock)

Bits	Reset Value	Name	Description
7..0	0x00	**V_PLL2_M**	multiplier M for PLL 2 (range 0 … 255)
13..8	0x00	**V_PLL2_P**	divider P for PLL 2 (range 0 … 63)
15..14	0	**V_PLL2_S**	divider S for PLL 2 (range 0 … 3)
25..16		**(reserved)**	
26	1	**V_PLL2_PWRDN**	sets the PLL 2 in power down mode '0' = power on '1' = power down
27		**(reserved)**	
28	0	**V_PLL2_SEL**	selects the PLL 2 '0' = inactive '1' = active
31..29		**(reserved)**	

R_DIV1_CFG	(read / write)	0x00080028

Configuration register for the predivider of the system clock generation

Bits	Reset Value	Name	Description
7..0	1	**V_CNT1A_M**	multiplier for the modulo-counter CNT 1A (range 0 … 255)
15..8	1	**V_CNT1A_N**	divider for the modulo-counter CNT 1A (range 1 … 255)
16	0	**V_PLL1_SEL**	selects the PLL 1 '0' = inactive '1' = active
17	0	**V_DIV1_SEL**	selects the divider DIV 1 '0' = inactive '1' = active
31..18		**(reserved)**	

R_OSC_CFG	(read / write)	0x00080038

Configuration register for the USB clock and the USB signal receiver

Bits	Reset Value	Name	Description
0	0	**V_OSC2_EN**	enables the oscillator OSC 2 '0' = off '1' = on
1		**(reserved)**	
2	1	**V_USB_SREC_OFF**	enables the USB pad single-ended receiver '0' = enabled '1' = disabled
3	1	**V_USB_OFF**	enables the USB pad differential receiver '0' = enabled '1' = disabled
7..4		**(reserved)**	

R_CNT1B_CFG	(read / write)	0x00080040

Configuration register for the f_{ext} clock generation with the modulo counter CNT 1B

Bits	Reset Value	Name	Description
7..0	0x01	**V_CNT1B_M**	multiplying value of the counter CNT 1B (range 1 … 255)
15..8	0x01	**V_CNT1B_N**	divider of the counter CNT 1B (range 1 … 255) **Note:** N must be greater or equal than M.

3.2 Timer modules

Table 12: *Overview of the* HFC-S active *timer pins (all primary function)*

Number	Name	Description
1	CARRY1	timer 1 carry signal
119	WDT	carry signal of the watchdog timer
159	PWM_OUT	PWM output
160	CARRY2	Timer 2 carry signal

Table 13: *Overview of the* HFC-S active *timer registers*

Address	Name	Page	Address	Name	Page
0x00090000	R_TIMER	50	0x00090024	R_GPIO_CTRL1	131
0x0009002C	R_TIMER_PRELD	50	0x00080000	R_FIQ_CTRL	57
0x00090004	R_TIMER_CFG1	51	0x00080004	R_IRQ_CTRL	58
0x00090008	R_WD	51	0x00080008	R_FIQ_STATUS	59
0x0009000C	R_PWM_CFG	52	0x0008000C	R_IRQ_STATUS	60
0x00090010	R_TIMER_CFG2	53			

The HFC-S active has 4 independent programable timers with interrupt capability:

- Timer 1 and Timer 2 for general usage,
- a PWM counter for a simple digital to analog conversion,
- and a watchdog timer.

The PWM counter can also be used as a timer for other purposes. A detailed description of the timers is given in the following sections.

3.2.1 Timer 1 and Timer 2

The HFC-S active has two independent programmable 16 bit timers with interrupt capabilities and 8 bit prescaler. The counters accumulate on the ARM7^{TM} system clock. The carry signals of the 16 bit timers can be mapped on the pins 1 and 160 (see tab. 12). Figure 11 illustrates the internal structure of the 16 bit Timer 1. Timer 1 and Timer 2 are constructed identically.

Each timer can be set to a 16 bit value by writing the value to the register R_TIMER. This value is decremented with the clock signal

$$f_{dec} = \begin{cases} \frac{f_{sys}}{\text{V\_TI1\_PREDIV}+1} & ; \quad Timer\,1 \\ \frac{f_{sys}}{\text{V\_TI2\_PREDIV}+1} & ; \quad Timer\,2 \end{cases} .$$

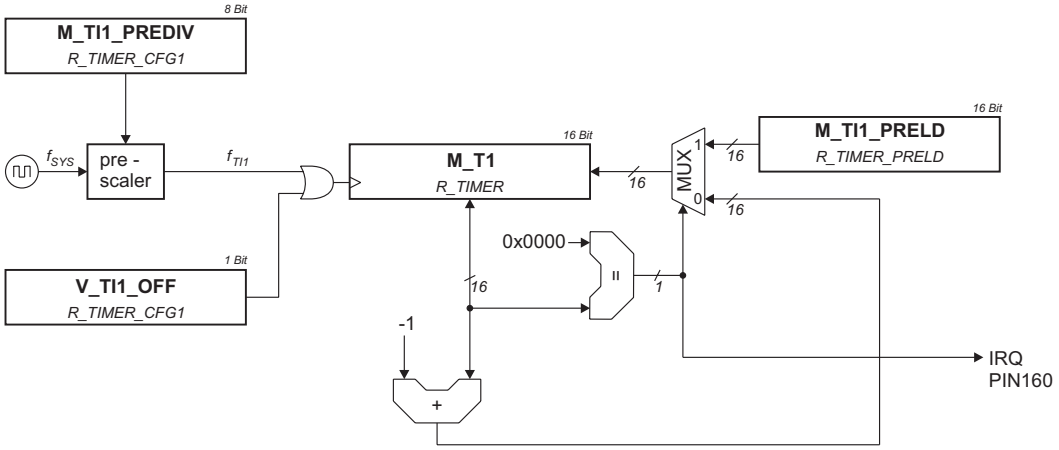


Figure 11: *Internal structure of the 16 bit Timer 1*

An Timer 1 interrupt occurs when V_TI1 (resp. V_TI2 for Timer 2) of the register R_TIMER reaches the value 0. Additionally, the Timer 1 zero-signal can be mapped to the pin 1 (resp. pin 160 for Timer 2) if V_GPIO0_TI1 (resp. V_GPIO1_TI2) of the register R_GPIO_CTRL1 is set to 1.

Reaching the value 0, the timer automatically reloads the predefined value of the register R_TIMER_PRELD. The periodically interrupt signal has the frequency

$$f_{IRQ} = \begin{cases} \frac{f_{dec}}{\text{V\_TI1\_PRELD}} = \frac{f_{sys}}{(\text{V\_TI1\_PREDIV}+1)\cdot\text{V\_TI1\_PRELD}} & ; \quad Timer\,1 \\ \frac{f_{dec}}{\text{V\_TI2\_PRELD}} = \frac{f_{sys}}{(\text{V\_TI2\_PREDIV}+1)\cdot\text{V\_TI2\_PRELD}} & ; \quad Timer\,2 \end{cases} .$$

The Timer 1 stops with V_TI1_OFF (resp. V_TI2_OFF for Timer 2) of the register R_TIMER_CFG1 is set to 1.

3.2.2 Watchdog timer

The 16 bit watchdog counter generates a global reset when elapsing. To prevent an HFC-S active reset the watchdog counter has to be reset by the software periodically. This can be carried out in two different ways. By setting the V_WD_RES bit in the R_TIMER_CFG1 register or by programming the watchdog counter value R_WD directly. For critical system operation the watchdog counter can generate a pre-reset interrupt signal to warn the software of the coming global reset. The threshold value of the pre-reset interrupt signal can be set by the V_WD_LEV bits in the register R_TIMER_CFG2.

3.2.3 PWM counter

The PWM counter can be used to produce a pulse width modulated signal for a simple digital to analog conversion. Therefore, the counter is periodically incremented. Reaching the value 0x0000 generates an interrupt, if enabled in the register R_TIMER_CFG2.

The pulse width can be set with a resolution of 10 bit (bitmap V_PWM_PWIDTH of the register R_PWM_CFG). The PWM signal is 0 for V_PWM $<$ V_PWM_PWIDTH and 1 otherwise. The generated PWM signal can be mapped on the GPIO[2] pin (see GPIO register R_GPIO_CTRL1).

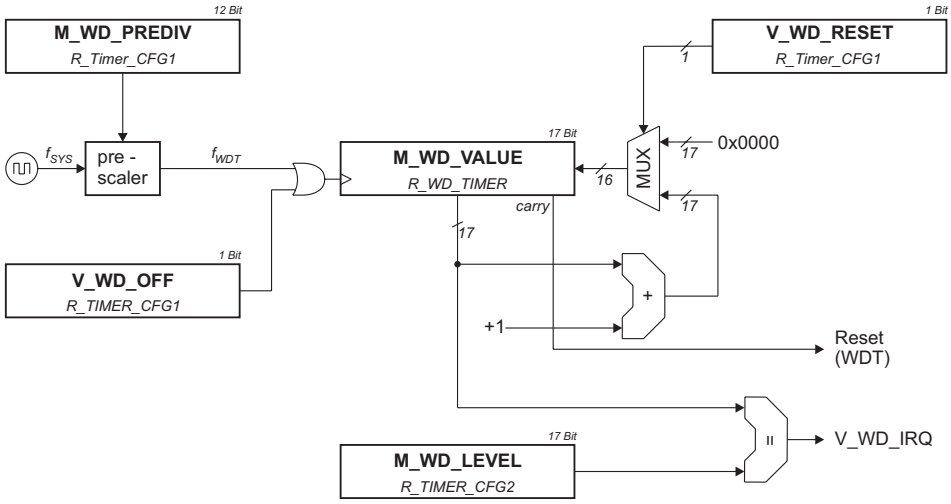


Figure 12: *Internal structure of the watchdog timer*

The PWM counter is clocked by the system clock f_{sys} and has a 12 bit prescaler (bitmap V_PWM_PREDIV of the register R_PWM_CFG) to control the counter frequency.

If the PWM counter isn't used for pulse width modulation, it can be used as a 10 bit counter with interrupt functionality. Therefore, the software has to write the initial value to the PWM counter register R_PWM_CFG in the interrupt service routine.

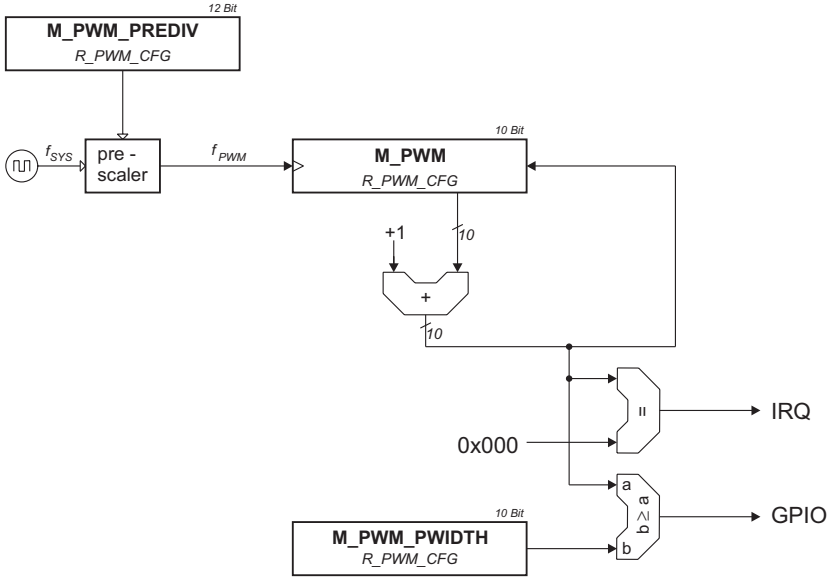


Figure 13: *Internal structure of the PWM counter*

3.2.4 Register description

R_TIMER	(read / write)	0x00090000

Counter register for timer 1 and timer 2

Bits	Reset Value	Name	Description
15..0	0xFFFF	**V_TI1**	counter value of timer 1 (count down counter)
31..16	0xFFFF	**V_TI2**	counter value of timer 2 (count down counter)

R_TIMER_PRELD	(read / write)	0x0009002C

Preload value register for timer 1 and timer 2

Bits	Reset Value	Name	Description
15..0	0x0000	**V_TI1_PRELD**	Preload value for timer 1. The timer is loaded with this preload value if it has counted to zero.
31..16	0x0000	**V_TI2_PRELD**	Preload value for timer 2. The timer is loaded with this preload value if it has counted to zero.

R_TIMER_CFG1	(read / write)	0x00090004

Timer control register: predivider for timer 1 and timer 2 and watchdog timer

Bits	Reset Value	Name	Description
7..0	0xBB	**V_TI1_PREDIV**	predivider value for timer 1
15..8	0x00	**V_TI2_PREDIV**	predivider value for timer 2
27..16	0x0BB	**V_WD_PREDIV**	predivider value for the watchdog timer
28	0	**V_TI1_OFF**	stop timer 1 '0' = run '1' = stop
29	0	**V_TI2_OFF**	stop timer 2 '0' = run '1' = stop
30	0	**V_WD_OFF**	stop the watchdog timer '0' = run '1' = stop
31	0	**V_WD_RES**	resets the watchdog timer to zero, this bit is set back automatically '0' = no reset '1' = reset

R_WD	(read / write)	0x00090008

Counter register for the watchdog timer

Bits	Reset Value	Name	Description
15..0	0x0000	**V_WD_VALUE**	Counter value of the watchdog timer (count-up counter) The carry signal of the watchdog counter generates the watchdog reset signal
31..16		**(reserved)**	

R_PWM_CFG	(read / write)	0x0009000C

PWM counter control register for PWM pulse generation.

The output signal can be mapped on the pins GPIO[0] and GPIO[1] (see primary GPIO description).

Bits	Reset Value	Name	Description
9..0	0x000	**V_PWM**	PWM counter value (count-up counter)
19..10	0x001	**V_PWM_PWIDTH**	sets the pulse width of the PWM counter
31..20	0x000	**V_PWM_PREDIV**	prescaler for the PWM counter

R_TIMER_CFG2	(read / write)	0x00090010

Timer interrupt status and control register

Bits	Reset Value	Name	Description
0	0	**V_TI1_IRQ**	shows the status for timer 1 interrupt (An interrupt is generated if the timer has counted to zero. Writing a zero to this bit resets the interrupt request.) '0' = no interrupt request '1' = interrupt request
1	0	**V_TI2_IRQ**	shows the status for timer 2 interrupt (An interrupt is generated if the timer has counted to zero. Writing a zero to this bit resets the interrupt request.) '0' = no interrupt request '1' = interrupt request
2	0	**V_WD_IRQ**	shows the status for the watchdog interrupt (Writing a zero to this bit resets the interrupt request.) '0' = no interrupt request '1' = interrupt request
3	0	**V_PWM_IRQ**	shows the status for PWM interrupt (Writing a zero to this bit resets the interrupt request.) '0' = no interrupt request '1' = interrupt request
4	0	**V_TI1_EN**	enables the timer 1 interrupt '1' = interrupt enable '0' = interrupt disable
5	0	**V_TI2_EN**	enables the timer 2 interrupt '1' = interrupt enable '0' = interrupt disable
6	0	**V_WD_EN**	enables the watchdog interrupt '1' = interrupt enable '0' = interrupt disable
7	0	**V_PWM_EN**	enables the PWM interrupt '1' = interrupt enable '0' = interrupt disable
23..8	0x8000	**V_WD_LEV**	Sets the threshold value for the generation of the interrupt signal. When the watchdog counter reaches the threshold value, a watchdog interrupt signal is generated to give the CPU the possibility to reset the watchdog counter in an interrupt service routine.
31..24		**(reserved)**	

3.3 Interrupt processing of the HFC-S active

Table 14: *Overview of the* HFC-S active *interrupt pins (all primary function)*

Number	Name	Description	Number	Name	Description
1	CARRY1	timer 1 carry signal	119	WDT	carry signal of the watch-dog timer
88	EOFT	EOFT signal of the S/T interface	120	PFS3	peripheral frame sync 3 signal with interrupt capability
89	DK_REP	DK_REP signal of the S/T interface	121	PFS2	peripheral frame sync 2 signal with interrupt capability
90	DK_EN	DK_EN signal of the S/T interface	122	PSF1	peripheral frame sync 1 signal with interrupt capability
115	CLK_ST	S/T clock f_{ISDN}	157	PFS0	peripheral frame sync 0 signal with interrupt capability
116	CLK_EXT	clock for external devices (f_{ext})	158		
117			159	PWM_OUT	PWM output
118	FSC_TE	FSC_TE signal of the S/T interface	160	CARRY2	Timer 2 carry signal

Table 15: *Overview of the* HFC-S active *interrupt registers*

Address	Name	Page	Address	Name	Page
0x00080000	R_FIQ_CTRL	57	0x00090028	R_FSC_IRQ	66
0x00080004	R_IRQ_CTRL	58	0x00090014	R_GPIO_CFG	129
0x00080008	R_FIQ_STATUS	59	0x00090018	R_GPIO_IRQ_CTRL	129
0x0008000C	R_IRQ_STATUS	60	0x000C000C	R_ST_B12_IRQ_STATUS	77
0x000E0004	R_USB_CFG	148	0x000C0014	R_ST_B12_IRQ_EN	78
0x000A002C	R_UART_IRQ_CFG	144	0x000C0018	R_ST_D_IRQ_EN	78
0x00090010	R_TIMER_CFG2	53			

3.3.1 Functional description

The ARM7^{TM} CPU has two levels of interrupt processing. One interrupt level is the fast interrupt (FIQ) and the other interrupt level is the normal interrupt (IRQ).

The HFC-S active has a hierarchically organized interrupt controlling system. Every module listed in 16 has an own interrupt controller (called sub-controller). Every module interrupt can be programmed as a FIQ (fast interrupt) or as an IRQ (normal interrupt). Figure 14 illustrates the interrupt architecture of the HFC-S active.

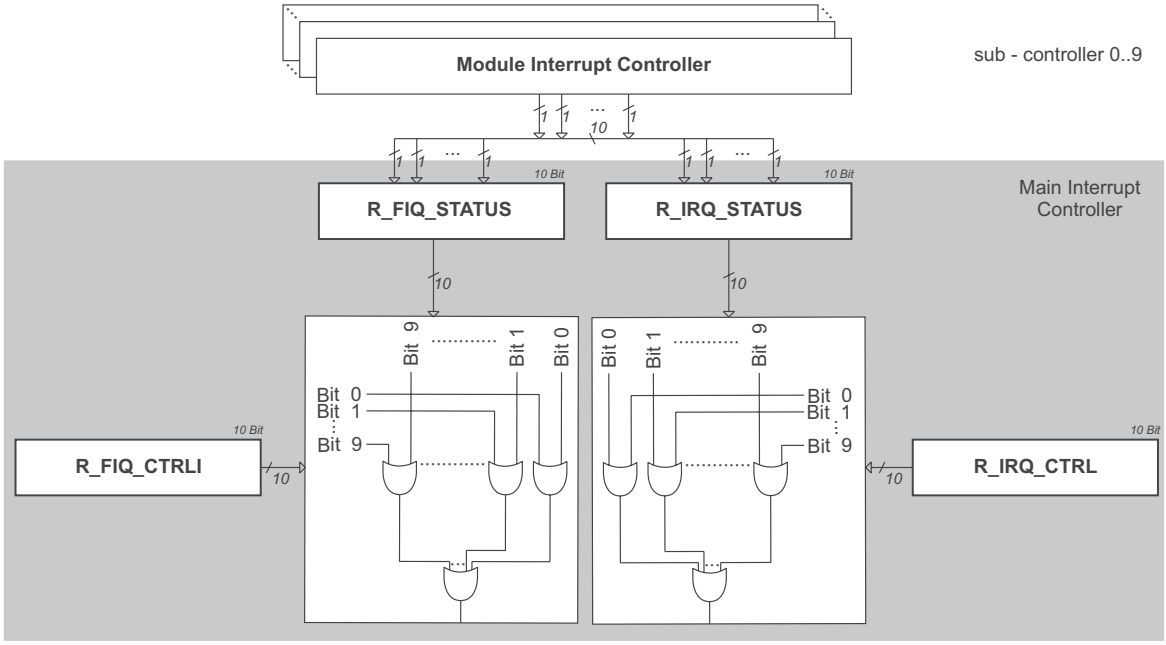


Figure 14: *Interrupt control structure of* HFC-S active

Table 16: *Bit numbering of the interrupt sub-controller*

IRQ / FIQ bit number	Sub-controller
0	USB module
1	UART module
2	Timer module
3	FSC PLL module
4	GPIO module
5	PFS 0 of FSC-PLL module,
6	PFS 1 of FSC-PLL module,
7	PFS 2 of FSC-PLL module,
8	PFS 3 of FSC-PLL module,
9	S/T-HDLC module

The main interrupt controller has 10 interrupt sources coming from the 10 sub-controllers. All interrupts are maskable in the main controller and in the sub-controllers. When an interrupt has occurred, the software must read the corresponding interrupt status register (fast interrupt or normal interrupt) of the main controller to detect the module which generated the interrupt request. After this the software has to read the interrupt status register of the corresponding module. The interrupt request must be set back by writing a zero value to the interrupt status register of the module.

Table 17: *Bit names of the interrupt registers*

IRQ / FIQ	Register			
bit number	R_FIQ_CTRL	R_IRQ_CTRL	R_FIQ_STATUS	R_IRQ_STATUS
0	V_FIQ_USB_EN	V_IRQ_USB_EN	V_FIQ_USB	V_IRQ_USB
1	V_FIQ_UART_EN	V_IRQ_UART_EN	V_FIQ_UART	V_IRQ_UART
2	V_FIQ_TI_EN	V_IRQ_TI_EN	V_FIQ_TI	V_IRQ_TI
3	V_FIQ_PLL_EN	V_IRQ_PLL_EN	V_FIQ_PLL	V_IRQ_PLL
4	V_FIQ_GPIO_EN	V_IRQ_GPIO_EN	V_FIQ_GPIO	V_IRQ_GPIO
5	V_FIQ_PFS0_EN	V_IRQ_PFS0_EN	V_FIQ_PFS0	V_IRQ_PFS0
6	V_FIQ_PFS1_EN	V_IRQ_PFS1_EN	V_FIQ_PFS1	V_IRQ_PFS1
7	V_FIQ_PFS2_EN	V_IRQ_PFS2_EN	V_FIQ_PFS2	V_IRQ_PFS2
8	V_FIQ_PFS3_EN	V_IRQ_PFS3_EN	V_FIQ_PFS3	V_IRQ_PFS3
9	V_FIQ_ST_EN	V_IRQ_ST_EN	V_FIQ_ST	V_IRQ_ST

3.3.2 Register description of the main interrupt controller

R_FIQ_CTRL	(read / write)	0x00080000

Interrupt control register to control the FIQ sources
'1' = enable
'0' = disable

Bits	Reset Value	Name	Description
0	0	**V_FIQ_USB_EN**	Enables the FIQ for the USB module
1	0	**V_FIQ_UART_EN**	Enables the FIQ for the UART module
2	0	**V_FIQ_TI_EN**	Enables the FIQ for the Timer module
3	0	**V_FIQ_PLL_EN**	Enables the FIQ for the FSC PLL module
4	0	**V_FIQ_GPIO_EN**	Enables the FIQ for the GPIO module
5	0	**V_FIQ_PFS0_EN**	Enables the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
6	0	**V_FIQ_PFS1_EN**	Enables the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
7	0	**V_FIQ_PFS2_EN**	Enables the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
8	0	**V_FIQ_PFS3_EN**	Enables the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
9	0	**V_FIQ_ST_EN**	Enables the FIQ for the S/T module
15..10		**(reserved)**	

(See table 17 on page 56 to identify the modules and bit names.)

R_IRQ_CTRL	(read / write)	0x00080004

Interrupt control register to control the IRQ sources
'1' = enable
'0' = disable

Bits	Reset Value	Name	Description
0	0	**V_IRQ_USB_EN**	Enables the IRQ for the USB module
1	0	**V_IRQ_UART_EN**	Enables the IRQ for the UART module
2	0	**V_IRQ_TI_EN**	Enables the IRQ for the Timer module
3	0	**V_IRQ_PLL_EN**	Enables the IRQ for the FSC PLL module
4	0	**V_IRQ_GPIO_EN**	Enables the IRQ for the GPIO module
5	0	**V_IRQ_PFS0_EN**	Enables the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
6	0	**V_IRQ_PFS1_EN**	Enables the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
7	0	**V_IRQ_PFS2_EN**	Enables the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
8	0	**V_IRQ_PFS3_EN**	Enables the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
9	0	**V_IRQ_ST_EN**	Enables the IRQ for the S/T module
15..10		**(reserved)**	

(See table 17 on page 56 to identify the modules and bit names.)

R_FIQ_STATUS	(read / write)	0x00080008

Interrupt status register for the FIQ sources
'1' = interrupt request
'0' = no interrupt request
The interrupt status register represents the status of each module. When the request of the corresponding module is set back by the software (in the module interrupt status register) the status entry is set back automatically.

The FIQ status is only shown if the corresponding module interrupt is enabled in the register R_FIQ_CTRL.

Bits	Reset Value	Name	Description
0	0	**V_FIQ_USB**	Status of the FIQ for the USB module
1	0	**V_FIQ_UART**	Status of the FIQ for the UART module
2	0	**V_FIQ_TI**	Status of the FIQ for the Timer module
3	0	**V_FIQ_PLL**	Status of the FIQ for the FSC PLL module
4	0	**V_FIQ_GPIO**	Status of the FIQ for the GPIO module
5	0	**V_FIQ_PFS0**	Status of the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
6	0	**V_FIQ_PFS1**	Status of the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
7	0	**V_FIQ_PFS2**	Status of the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
8	0	**V_FIQ_PFS3**	Status of the FIQ for the peripheral frame sync signal (PCM highway-Interface) module
9	0	**V_FIQ_ST**	Status of the FIQ for the S/T module
15..10		**(reserved)**	

(See table 17 on page 56 to identify the modules and bit names.)

R_IRQ_STATUS	(read / write)	0x0008000C

Interrupt status register for the IRQ sources
'1' = interrupt request
'0' = no interrupt request
The interrupt status register represents the status of each module. If the request of the corresponding module is set back by the software (in the module interrupt status register) the status entry is set back automatically.

The IRQ status is only shown if the corresponding module interrupt is enabled in the register R_IRQ_CTRL.

Bits	Reset Value	Name	Description
0	0	**V_IRQ_USB**	Status of the IRQ for the USB module
1	0	**V_IRQ_UART**	Status of the IRQ for the UART module
2	0	**V_IRQ_TI**	Status of the IRQ for the Timer module
3	0	**V_IRQ_PLL**	Status of the IRQ for the FSC PLL module
4	0	**V_IRQ_GPIO**	Status of the IRQ for the GPIO module
5	0	**V_IRQ_PFS0**	Status of the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
6	0	**V_IRQ_PFS1**	Status of the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
7	0	**V_IRQ_PFS2**	Status of the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
8	0	**V_IRQ_PFS3**	Status of the IRQ for the peripheral frame sync signal (PCM highway-Interface) module
9	0	**V_IRQ_ST**	Status of the IRQ for the S/T module
15..10		**(reserved)**	

(See table 17 on page 56 to identify the modules and bit names.)

4 ISDN related modules

4.1 FSC-PLL module

Table 18: *Overview of the* HFC-S active *FSC-PLL pins (primary function pins marked with *)*

Number	Name	Description
96	FSC0	frame sync signal for PCM highway 1
97	PFS3	peripheral frame sync signal
98	PFS2	peripheral frame sync signal
99	PFS1	peripheral frame sync signal
100	PFS0	peripheral frame sync signal
104	FSC1	frame sync signal for PCM highway 2
110	FSC2	frame sync signal for PCM highway 3
117 *		
118 *	FSC_TE	FSC_TE signal of the S/T interface
120 *	PFS3	peripheral frame sync 3 signal with interrupt capability
121 *	PFS2	peripheral frame sync 2 signal with interrupt capability
122 *	PSF1	peripheral frame sync 1 signal with interrupt capability
157 *	PFS0	peripheral frame sync 0 signal with interrupt capability
158 *		

Table 19: *Overview of the* HFC-S active *FSC-PLL registers*

Address	Name	Page	Address	Name	Page
0x00080000	R_FIQ_CTRL	57	0x00090034	R_FSC_CONST	68
0x00080004	R_IRQ_CTRL	58	0x000B0204	R_HW1_CTRL	111
0x00080008	R_FIQ_STATUS	59	0x000B0208	R_HW2_CTRL	113
0x0008000C	R_IRQ_STATUS	60	0x000B020C	R_HW3_CTRL	115
0x00080030	R_GPIO_CTRL2	133	0x000B01F4	R_PFS0_CFG	108
0x00090014	R_GPIO_CFG	129	0x000B01F6	R_PFS1_CFG	108
0x00090024	R_GPIO_CTRL1	131	0x000B01F8	R_PFS2_CFG	109
0x00090028	R_FSC_IRQ	66	0x000B01FA	R_PFS3_CFG	109
0x00090030	R_FSC_CFG	67			

The FSC-PLL[5] is one of the central modules of the HFC-S active. It is responsible for the generation of the ISDN frame synchronization pulse. The FSC signal is the central clock of 8 kHz for ISDN telephone network, generated by the central office. The ISDN controller, the PCM highways and the CODEC interface work on this synchronization signal.

The main task of the FSC-PLL is to eliminate jitter of the external frame synchronization clock by

[5] FSC: frame synchronization clock

means of the higher resolution of the system clock.

4.1.1 FSC source selection

The source signal for the synchronization clock is selectable by the software. If no FSC clock is generated from a source module (PCM highway, S/T-HDLC or a peripheral device connected to GPIO0 ... GPIO15), the FSC-PLL is free-running.

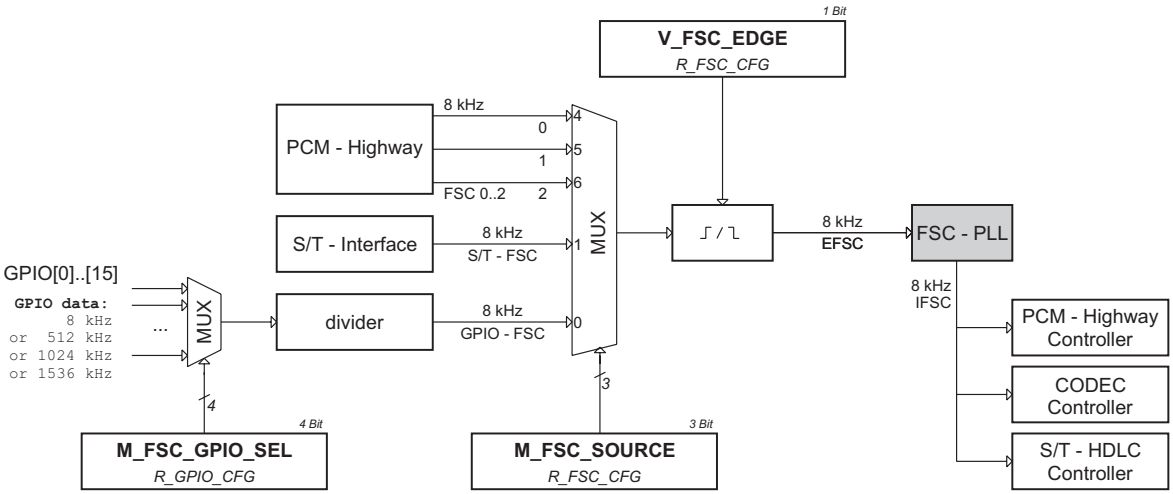


Figure 15: *Overview of the FSC signal source selection of the FSC-PLL*

Figure 15 illustrates the FSC source selection. The bitmap V_FSC_SRC of the register R_FSC_CFG offers three possibilities for the FSC source:

1. FSC0, FSC1 or FSC2 of the PCM highway,
2. FSC_TE of the S/T interface,
3. and one GPIO pin out of GPIO0... GPIO15.

The GPIO pin can be selected with the bitmap V_FSC_GPIO_SEL of the register R_GPIO_CFG. As the GPIO synchronization signal f_{in} can be a multiple of 8 kHz, a programmable 10 bit divider must generate the required signal frequency f_{out}, i.e.

$$\text{V\_FSC\_PREDIV} = \frac{f_{in}}{f_{out}} - 1$$

where V_FSC_PREDIV is a bitmap of the register R_GPIO_CFG and $f_{out} = 8\,\text{kHz}$. In contrast to this, the S/T interface and the PCM highway work always on a 8 kHz synchronization signal.

The selected signal is called EFSC (external frame sync clock). The FSC-PLL accepts this signal and generates the internal frame syncronization clock (IFSC) which is used from the PCM highway, the CODECs and the S/T controller.

4.1.2 Functional description of the FSC-PLL

The FSC-PLL generates the internal FSC from the external FSC which is passed to the PCM highway controller, the CODEC interface and the S/T controller. The FSC-PLL module offers a very variable adjustment. Figure 16 illustrates the configuration possibilities. Three main parts perform the FSC-PLL functionality – the PLL counter, the PLL phase alignment and the FSC adjustment – and are described in the following paragraphs.

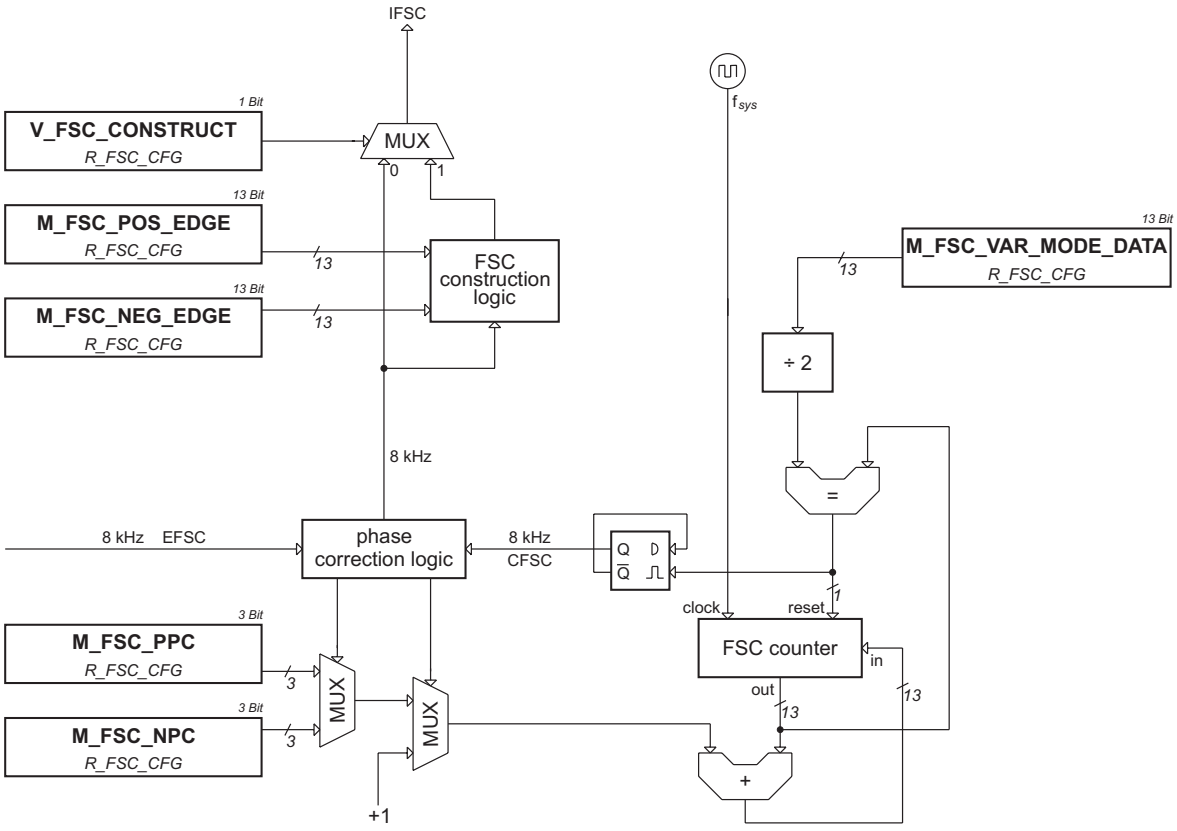


Figure 16: *Overview of the internal structure of the FSC-PLL*

PLL counter

The FSC-PLL counter is a 13 bit count-up counter which is clocked by f_{sys}. To generate the CFSC signal (see figure 16), the PSC counter must receive its reset signal every 125 μs. Therefore the counter end value has to be programmed by

$$\text{V\_FSC\_VARMODE\_DATA} = 1536 \cdot \frac{f_{sys}}{12.288\,\text{MHz}}$$

in the register R_FSC_CFG. This value must not be greater than 8191.

Furtheron, it is necessary to set (reserved) = 1 in the register R_FSC_CFG.

With each counter clock, the value of the FSC-PLL counter is incremented with a value which depends on the phase detection result which is described in the next paragraph. Reaching its end value, the counter restarts with zero.

PLL phase alignment

The FSC-PLL phase detection has two input signals of 8 kHz. The CFSC has to syncronize with EFSC, and in addition to this, it must suppress the EFSC jitter.

If no phase shift is needed, the FSC-PLL counter will always be incremented by 1. A phase offset between EFSC and CFSC causes a different increment with a positive or a negative value. These values can be programmed in the range 0 ... 7 with the bitmaps V_FSC_PPC resp. V_FSC_NPC (two complement) in the register R_FSC_CFG.

The phase correction may be performed with every pulse of the FSC or only every nth pulse. This is put into action with the bitmap V_FSC_CORT of the register R_FSC_CFG. Only the rising edge of the FSC is involved with the phase correction.

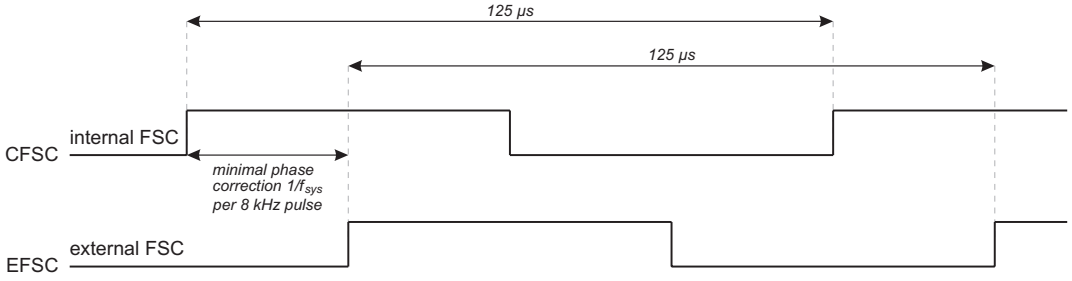


Figure 17: *Principle of the phase correction*

If a phase deviation is recognized, the FSC-PLL corrects the phase difference in the corresponding direction. The phase correction is carried out in both directions. The maximum phase jump per clock period (125 μs) is programmable. The resolution of the phase jump depends on the clock frequency f_{sys}. The minimum phase jump per 125 μs is 16.27 ns (at $f_{sys} = 61.44\,\mathrm{MHz}$).

The FSC-PLL module generates an interrupt (if enabled) every 125 μs. The interrupt source can be the internal FSC signal generated by the FSC-PLL module or / and the four peripheral FSC signals (see R_FSC_IRQ description).

4.1.3 The constructed FSC

The FSC-PLL has a constant phase shift (delay) of three system clock periods with reference to the external FSC signal due to the synchronization logic. In case that the constant phase delay leads to disadvantages for the external devices (e.g. PCM CODECs), the FSC-PLL module offers a programmable phase position.

Within the contruction logic block, the phase position of the rising and falling edges are independently programmable with a resolution of 13 bit each (bitmaps V_FSC_POS_EDGE and V_FSC_NEG_EDGE of the register R_FSC_CONST) like shown in figure 18.

The periodically FSC contruction starts with the rising edge of the FSC output from the phase correction logic block (called PLL-FSC). A 13 bit counter is incremented with f_{sys} clock beginning at zero. Simultaneously the constructed FSC is put to `high`.

If the counter reaches the value of V_FSC_NEG_EDGE (register R_FSC_CONST), the constructed FSC changes to `low`. A counter value equal to V_FSC_POS_EDGE (same register) sets the signal back to `high`. If this condition is not reached before the next rising edge of the PLL-FSC, this is

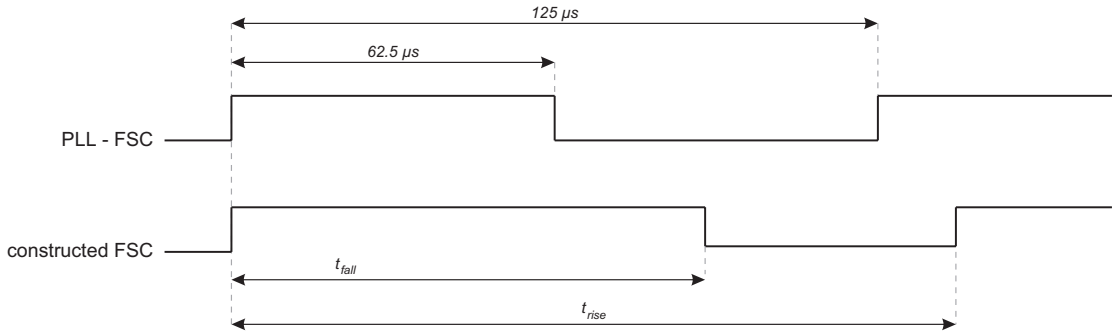


Figure 18: *Programmable phase position for the FSC signal*

done as well.

A neutral constructed FSC which is identical with the PLL-FSC is achieved with

$$\text{V\_FSC\_NEG\_EDGE} = 62.5\,\mu\text{s} \cdot f_{sys}$$

for a falling edge after 62.5 μs and

$$\text{V\_FSC\_POS\_EDGE} = 2 \cdot \text{V\_FSC\_NEG\_EDGE}$$

for a rising edge after 125 μs. Note, that the rising edge is not influenced with greater values or if it is zero.

A falling edge shift Δt_{fall} is configured with

$$\text{V\_FSC\_NEG\_EDGE} = (\Delta t_{fall} + 62.5\,\mu\text{s}) \cdot f_{sys}$$

where $-62.5\,\mu\text{s} \leq \Delta t_{fall} \geq 62.5\,\mu\text{s}$. If the neutral rising edge is located at $t = 125\,\mu$s, the shift Δt_{rise} needs a bitmap value

$$\text{V\_FSC\_POS\_EDGE} = (\Delta t_{rise} + 125\,\mu\text{s}) \cdot f_{sys}$$

with $-125\,\mu\text{s} \leq \Delta t_{rise} \leq 0$. The neutral rising edge can also be seen at $t = 0$. Then

$$\text{V\_FSC\_POS\_EDGE} = \Delta t_{rise} \cdot f_{sys}$$

within the range $0 \leq \Delta t_{rise} \leq 125\,\mu$s. Finally, an inverted constructed FSC is achieved with

$$\text{V\_FSC\_NEG\_EDGE} = 0$$

and

$$\text{V\_FSC\_POS\_EDGE} = 62.5\,\mu\text{s} \cdot f_{sys}\ .$$

4.1.4 Register description

R_FSC_IRQ	(read / write)	0x00090028

Interrupt status and enable register for the FSC interrupts

enable bits 0 . . . 4: '0' = interrupt disable, '1' = interrupt enable
status bits 6 . . . 10: '1' = interrupt, '0' = no interrupt
(Writing '0' sets back the interrupt request)

Bits	Reset Value	Name	Description
0	0	**V_FSC_IRQ**	interrupt enable register for the internal FSC signal
1	0	**V_PFSC0_IRQ**	interrupt enable register for the internal peripheral FSC signal PFS0
2	0	**V_PFSC1_IRQ**	interrupt enable register for the peripheral FSC signal PFS1
3	0	**V_PFSC2_IRQ**	interrupt enable register for the peripheral FSC signal PFS2
4	0	**V_PFSC3_IRQ**	interrupt enable register for the peripheral FSC signal PFS3
5		**(reserved)**	
6	0	**V_FSC_IRQ_STATUS**	interrupt status register for the internal FSC signal. The interrupt request is set back with '0'.
7	0	**V_PFSC0_IRQ_STATUS**	interrupt status register for the internal peripheral FSC signal (PFS0)
8	0	**V_PFSC1_IRQ_STATUS**	interrupt status register for the internal peripheral FSC signal (PFS1)
9	0	**V_PFSC2_IRQ_STATUS**	interrupt status register for the internal peripheral FSC signal (PFS2)
10	0	**V_PFSC3_IRQ_STATUS**	interrupt status register for the internal peripheral FSC signal (PFS3)
15..11		**(reserved)**	

R_FSC_CFG	(read / write)	0x00090030

FSC-PLL configuration register for programmable phase positioning of the internal FSC signal

Bits	Reset Value	Name	Description
2..0	0	**(reserved)**	
5..3	2	**V_FSC_PPC**	sets the phase jump for the positive phase correction '0' = no phase correction '1' = phase correction 1 clock cycle ... '7' = phase correction 7 clock cycle
8..6	7	**V_FSC_NPC**	sets the phase jump for the negative phase correction (two complement representation: '000' = 0, '001' = -7, '010' = -6, ..., '111' = -1)
9	0	**V_FSC_EDGE**	sets clock edge for synchronization '0'=falling '1'= rising
12..10	0	**V_FSC_CORT**	sets the number of frame sync pulses in with a phase correction is carried out '0' = the phase correction is done within every 125 μs '1' = the phase correction is done within every 250 μs ... '7' = the phase correction is done within every 1000 μs
15..13	0	**V_FSC_SRC**	selects the source signal for the FSC-PLL '000' = source is GPIO '001' = source is S/T module '100' = source is FSC0 (PCM highway) '101' = source is FSC1 (PCM highway) '110' = source is FSC2 (PCM highway) **Note:** other values deactivate the source
16	0	**V_FSC_CONST**	enables the variable phase positioning of the internal FSC signal '0' = off '1' = on
29..17	0x600	**V_FSC_VARMODE_DATA**	Prescaler for the FSC signal in variable mode of the FSC signal generation. In this mode the FSC-PLL can be adjust to every system programmed system frequency.
30		**(reserved)**	
31	0	**(reserved)**	must be set to '1'

R_FSC_CONST	(read / write)	0x00090034

FSC construct register for programmable phase position.

Bits	Reset Value	Name	Description
12..0	0x0000	**V_FSC_POS_EDGE**	value for rising clock position '1111000000000' = 7680 for 61.440 MHz '1100000000000' = 6144 for 49.152 MHz '1001000000000' = 4608 for 36.864 MHz '0110000000000' = 3072 for 24.576 MHz '0011000000000' = 1536 for 12.288 MHz
25..13	0x0000	**V_FSC_NEG_EDGE**	value for falling clock position '1111000000000' = 7680 for 61.440 MHz '1100000000000' = 6144 for 49.152 MHz '1001000000000' = 4608 for 36.864 MHz '0110000000000' = 3072 for 24.576 MHz '0011000000000' = 1536 for 12.288 MHz
31..26		**(reserved)**	

4.2 S/T-HDLC controller

Table 20: *Overview of the* HFC-S active *S/T-HDLC pins*

Number	Name	Description	Number	Name	Description
76	TX1_HI	transmit port (high) for the S/T interface	81	R2	receive port for the S/T interface
77	/TX_EN	transmit enable port	82	LEV_R2	level detect for R2
78	TX2_HI	transmit port (high) for the S/T interface	83	LEV_R1	Level detect for R1
79	TX2_LO	transmit port (low) for the S/T interface	84	R1	receive port for the S/T interface
80	TX1_LO	transmit port (low) for the S/T interface	85	ADJ_LEV	adjust level control for the S/T interface

Table 21: *Overview of the* HFC-S active *S/T-HDLC registers*

Address	Name	Page	Address	Name	Page
0x000C0000	R_ST_CFG	72	0x000C0048	R_ST_B12_STATUS	84
0x000C0004	R_ST_TX_STATUS	74	0x000C004C	R_ST_D_STATUS	84
0x000C0008	R_ST_RX_STATUS	74	0x000C00C0	R_ST_WR_STATES	86
0x000C000C	R_ST_B12_IRQ_STATUS	77	0x000C00C0	R_ST_RD_STATES	85
0x000C0010	R_ST_D_FIFO_STATUS	78	0x000C00C4	R_ST_CTRL1	87
0x000C0014	R_ST_B12_IRQ_EN	78	0x000C00C8	R_ST_CTRL2	88
0x000C0018	R_ST_D_IRQ_EN	78	0x000C00CC	R_ST_CTRL3	88
0x000C001C	R_ST_B1_TX_FIFO	79	0x000C00D0	R_ST_SQ_MF	89
0x000C0020	R_ST_B2_TX_FIFO	79	0x000C00DC	R_ST_CLK_CTRL	89
0x000C0024	R_ST_D_TX_FIFO	80	0x000C00F0	R_ST_B1_RX	90
0x000C0028	R_ST_B1_RX_FIFO	80	0x000C00F4	R_ST_B1_TX	90
0x000C0030	R_ST_B2_RX_FIFO	81	0x000C00F8	R_ST_B2_RX	90
0x000C0034	R_ST_D_RX_FIFO	81	0x000C00FC	R_ST_B2_TX	90
0x000C0038	R_ST_B1_CRC	81	0x000C0100	R_ST_D_RX	91
0x000C003C	R_ST_B2_CRC	82	0x000C0104	R_ST_D_TX	91
0x000C0040	R_ST_D_CRC	82	0x000C0108	R_ST_E_RX	91
0x000C0044	R_ST_CTRL	83			

4.2.1 Functional description

The S/T-HDLC module is an *ISDN S/T-HDLC Basic Rate Interface* (BRI).

The S/T-HDLC interface and the PCM highway form a functional unit for ISDN telecommunication applications. Additionally, the CODEC interface can be involved in the data flow via the ARM7^{TM} CPU.

Various options of the S/T-HDLC module allow a very flexible use of the module. The integrated HDLC controller for D-, B1- and B2-channel permits the construction of HDLC frames with an arbitrary length. The HDLC-frames of the B1- and B2-channel can be sent to the S/T interface or to the switching unit.

The S/T-HDLC module supports 32 bit data access only, that means always 4 byte are processed together. The clock frequency is fixed to 12.288 MHz.

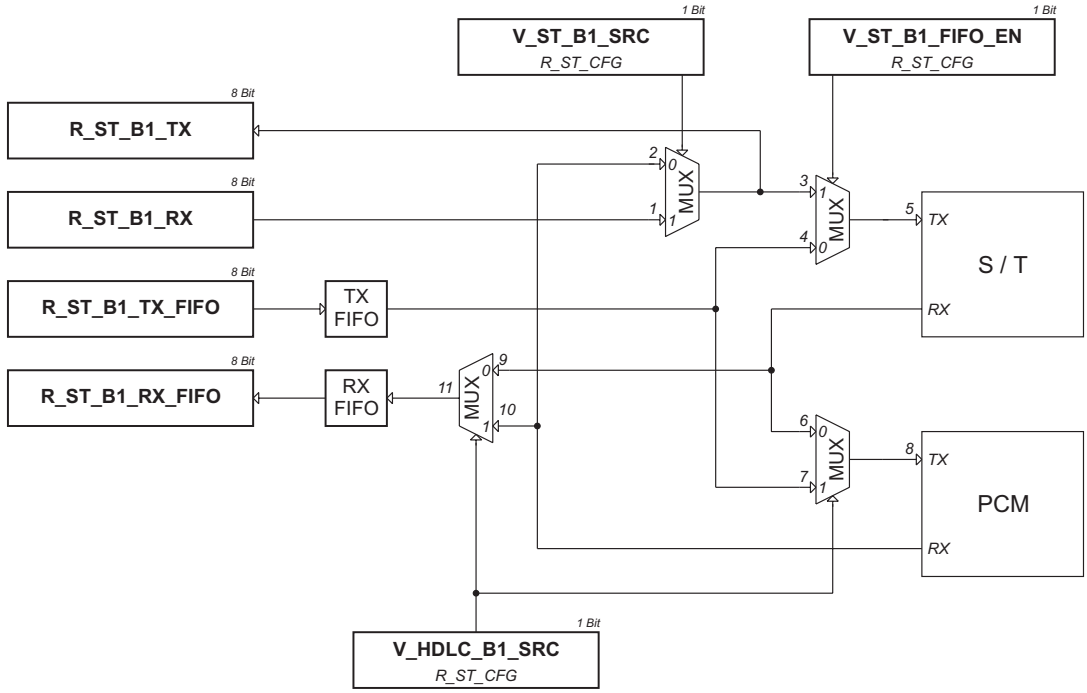


Figure 19: *Data path configuration options for the S/T-HDLC module (only shown for the B1-channel)*

Figure 19 illustrates the configuration options of the S/T-HDLC module. Data sources and destinations are selectable from

- the ARM7^{TM} CPU (directly or via FIFO),
- the PCM highway (via switching unit),
- or the S/T interface (via switching unit).

It is also possible to transmit data to the S/T interface without FIFO buffering. In this case the CPU has to ensure the timing constraints.

For the CRC generation the standard check sum CCITT-16 ($x^{16} + x^{12} + x^5 + 1$) is used. Figure 20 illustrates the composition of a HDLC frame.

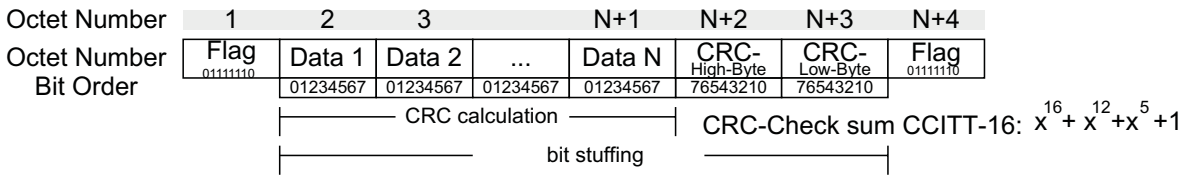


Figure 20: *HDLC frame format*

Table 22: *Control field organization of the HDLC mode*

Bit	Transmit	Receive
0	Start HDLC frame with byte 0	Byte 0 is beginning of the HDLC frame
1	Start HDLC frame with byte 1	Byte 1 is beginning of the HDLC frame
2	Start HDLC frame with byte 2	Byte 2 is beginning of the HDLC frame
3	reserved	reserved
4	Stop HDLC frame with byte 0	Byte 0 is end of the HDLC frame
5	Stop HDLC frame with byte 1	Byte 1 is end of the HDLC frame
6	Stop HDLC frame with byte 2	Byte 2 is end of the HDLC frame
7	reserved	CRC of the HDLC frame is correct

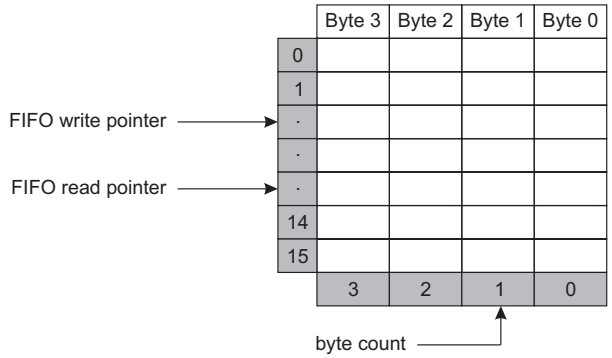


Figure 21: *FIFO pointer structure*

The HDLC controller of the S/T-HDLC interface has a 64 byte FIFO for each channel (B1, B2 and D) and each direction. The FIFOs can be used in transparent mode and in HDLC mode. They are organized in blocks of 16 x 32 bit. In HDLC mode the fourth byte of the FIFO is interpreted as a control field for data flow controlling. Table 22 illustrates the control field structure for transmit and receive direction.

Figure 21 shows the pointer structure of the FIFO.

4.2.2 Register description

R_ST_CFG	(read / write)	0x000C0000
S/T-HDLC configuration register		

Bits	Reset Value	Name	Description
0	0	**V_ST_B1_SRC**	selects the source for the data on the B1-channel '0' = ARM7^{TM} CPU '1' = PCM highway
1	0	**V_ST_B2_SRC**	selects the source for the data on the B2-channel '0' = ARM7^{TM} CPU '1' = PCM highway
3..2	0	**V_ST_B1_TX_MODE**	sets the sending data mode for the B1-channel '00' = transparent mode '10' = HDLC mode without CRC generation '11' = HDLC mode with CRC-16 generation
5..4	0	**V_ST_B1_RX_MODE**	sets the receiving data mode for the B1-channel '00' = transparent mode '10' = HDLC mode without CRC generation '11' = HDLC mode with CRC-16 generation
6	1	**V_ST_B1_RX_FIFO_STATUS**	resets the B1 receive FIFO pointer
7	0	**V_ST_B1_TX_MSB**	sets the bit direction for the sending data in the B1-channel '0' = LSB first '1' = MSB first
8	1	**V_ST_B1_RX_MSB**	sets the bit direction for the receiving data in the B1-channel '0' = LSB first '1' = MSB first
9	1	**V_ST_B1_RX_LSB**	sets the format for the raw data in the B1 receive FIFO '0' = original (MSB first) '1' = mirrored (LSB first)
11..10	0	**V_ST_B2_TX_MODE**	sets the sending data mode for the B2-channel '00' = transparent mode '10' = HDLC mode without CRC generation '11' = HDLC mode with CRC-16 generation
13..12	0	**V_ST_B2_RX_MODE**	sets the receiving data mode for the B2-channel '00' = transparent mode '10' = HDLC mode without CRC generation '11' = HDLC mode with CRC-16 generation
14	1	**V_ST_B2_RX_FIFO_STATUS**	resets the B2 receive FIFO pointer
15	0	**V_ST_B2_TX_MSB**	sets the bit direction for the sending data in the B2-channel '0' = LSB first '1' = MSB first

Bits	Reset Value	Name	Description
16	1	**V_ST_B2_RX_MSB**	sets the bit direction for the receiving data in the B2-channel '0' = LSB first '1' = MSB first
17	1	**V_ST_B2_RX_FIFO_LSB**	sets the format for the raw data in the B2 receive FIFO '0' = original (MSB first) '1' = mirrored (LSB)
19..18	0	**V_ST_D_TX_MODE**	sets the sending data mode for the D-channel '00' = transparent mode '10' = HDLC mode without CRC generation '11' = HDLC mode with CRC-16 generation
21..20	0	**V_ST_D_RX_MODE**	sets the receiving data mode for the D-channel '00' = transparent mode '10' = HDLC mode no CRC generation '11' = HDLC mode with CRC-16 generation
22	1	**V_ST_D_RX_FIFO_STATUS**	resets the D receive FIFO pointer
23	0	**V_ST_D_TX_MSB**	sets the bit direction for the sending data on the B2-channel '0' = LSB first '1' = MSB first
24	1	**V_ST_D_RX_MSB**	sets the bit direction for the receiving data on the B2-channel '0' = LSB first '1' = MSB first
25	1	**V_ST_D_RX_FIFO_LSB**	sets the format for the raw data in the B2 receive FIFO '0' = original (MSB) '1' = mirrored (LSB)
26	1	**V_ST_B1_FIFO_EN**	activates the sending FIFO for the B1-channel '0' = inactive '1' = active
27	1	**V_ST_B2_FIFO_EN**	activates the sending FIFO for the B2-channel '0' = inactive '1' = active
28	1	**V_ST_D_FIFO_EN**	activates the sending FIFO for the D-channel '0' = inactive '1' = active
29	1	**V_ST_LOOP**	sets all FIFOs in internal loop mode (for test only)
30	0	**V_HDLC_B1_SRC**	B1 receives data from S/T or PCM interface '0' = S/T '1' = PCM interface
31	0	**V_HDLC_B2_SRC**	B2 receives data from S/T or PCM interface '0' = S/T '1' = PCM interface

R_ST_TX_STATUS	(read only)	0x000C0004

FIFO status register for transmit channels

Bits	Reset Value	Name	Description
3..0	0	**V_B1_TX_FIFO_WRPTR**	displays FIFO write pointer of B1-channel transmitter
7..4	0	**V_B1_TX_FIFO_RDPTR**	displays FIFO read pointer of B1-channel transmitter
11..8	0	**V_B2_TX_FIFO_WRPTR**	displays FIFO write pointer of B2-channel transmitter
15..12	0	**V_B2_TX_FIFO_RDPTR**	displays FIFO read pointer of B2-channel transmitter
19..16	0	**V_D_TX_FIFO_WRPTR**	displays FIFO write pointer of D-channel transmitter
23..20	0	**V_D_TX_FIFO_RDPTR**	displays FIFO read pointer of D-channel transmitter
25..24	0	**V_B1_TX_CNT**	displays FIFO byte counter of B1-channel transmitter
27..26	0	**V_B2_TX_CNT**	displays FIFO byte counter of B2-channel transmitter
29..28	0	**V_D_TX_CNT**	displays FIFO byte counter of D-channel transmitter
31..30		**(reserved)**	

R_ST_RX_STATUS	(read only)	0x000C0008

FIFO status register for receive channels

Bits	Reset Value	Name	Description
3..0	0	**V_B1_RX_FIFO_WRPTR**	displays FIFO write pointer of B1-channel receiver
7..4	0	**V_B1_RX_FIFO_RDPTR**	displays FIFO read pointer of B1-channel receiver
11..8	0	**V_B2_RX_FIFO_WRPTR**	displays FIFO write pointer of B2-channel receiver
15..12	0	**V_B2_RX_FIFO_RDPTR**	displays FIFO read pointer of B2-channel receiver
19..16	0	**V_D_RX_FIFO_WRPTR**	displays FIFO write pointer of D-channel receiver
23..20	0	**V_D_RX_FIFO_RDPTR**	displays FIFO read pointer of D-channel receiver
25..24	0	**V_B1_RX_CNT**	displays FIFO byte counter of B1-channel receiver
27..26	0	**V_B2_RX_CNT**	displays FIFO byte counter of B2-channel receiver
29..28	0	**V_D_TX_CNT**	displays FIFO byte counter of D-channel receiver
31..30		**(reserved)**	

Table 23: *Bitmap description of the FIFO transmit status*

Bit number	Bit name	Description
0	empty	indicates that the transmit FIFO is empty
1	full	indicates that the transmit FIFO is full
2	overflow	indicates an overflow in the transmit FIFO (data has not been sent)
3	underflow	indicates an underflow in the transmit FIFO (in transparent mode: old data has been sent, in HDLC mode: 0xFF has been sent)
4	abort	indicates abort of HDLC frame in transmit channel
5	HDLC_mode	indicates that the transmitted data is an HDLC frame

Table 24: *Bitmap description of the FIFO receive status (B1- and B2-channel)*

Bit number	Bit name	Description
0	empty	indicates that the receive FIFO is empty
1	full	indicates that the receive FIFO is full
2	overflow	indicates an overflow in the receive FIFO (data has not been written into the FIFO)
3	underflow	indicates an underflow in the receive FIFO (in transparent mode: old data has been written, in HDLC mode: 0xFF has been written)
4	abort	indicates abort of HDLC frame in receive channel
5	CRC_OK	indicates correct CRC checksum of receive channel
6	HDLC_mode	indicates that a HDLC frame was received
7	align_error	indicates an error in the HDLC data stream of the receive channel (data is not byte aligned)
8	lost_error	indicates that data has been lost in receive channel

Table 25: *Bitmap description of the FIFO receive status (D-channel)*

Bit number	Bit name	Description
0	empty	indicates that the receive FIFO is empty
1	full	indicates that the receive FIFO is full
2	overflow	indicates an overflow in the receive FIFO (data has not been written into the FIFO)
3	underflow	indicates an underflow in the receive FIFO (in transparent mode: old data has been written, in HDLC mode: 0xFF has been written)
4	abort	indicates abort of HDLC frame in receive channel
5	CRC_OK	indicates correct CRC checksum of receive channel
6	HDLC_mode	indicates that a HDLC frame was received
7	align_error	indicates an error in the HDLC data stream of the receive channel (data is not byte aligned)
8	repeat	indicates a request to repeat the last HDLC frame in receive channel
9	lost_error	indicates that data has been lost in receive channel

R_ST_B12_IRQ_STATUS	**(read / write)**	0x000C000C
Interrupt status register for S/T-HDLC B1- and B2-channels		

Bits	Reset Value	Name	Description
5..0	0	**V_B1_TX_FIFO_STATUS**	indicates the B1-channel transmit FIFO status
14..6	0	**V_B1_RX_FIFO_STATUS**	indicates the B1-channel receive FIFO status
15		**(reserved)**	
21..16	0	**V_B2_TX_FIFO_STATUS**	indicates the B2-channel transmit FIFO status
30..22	0	**V_B2_RX_FIFO_STATUS**	indicates the B1-channel transmit FIFO status
31		**(reserved)**	

(see table 23 and 24 for bitmap explanation)

R_ST_D_FIFO_STATUS	(read / write)	0x000C0010

Interrupt status register for S/T-HDLC D-channel

Bits	Reset Value	Name	Description
5..0	0	**V_D_TX_FIFO_STATUS**	indicates the D-channel transmit FIFO status
15..6	0	**V_D_RX_FIFO_STATUS**	indicates the D-channel receive FIFO status

(see table 23 and 25 for bitmap explanation)

R_ST_B12_IRQ_EN	(read / write)	0x000C0014

Interrupt enable register for S/T-HDLC B1- and B2-channels
'0' = disable interrupt
'1' = enable interrupt

Bits	Reset Value	Name	Description
5..0	0	**V_B1_TX_IRQ_EN**	enables the B1 transmit interrupts
14..6	0	**V_B1_RX_IRQ_EN**	enables the B1 receive interrupts
15		**(reserved)**	
21..16	0	**V_B2_TX_IRQ_EN**	enables the B2 transmit interrupts
30..22	0	**V_B2_RX_IRQ_EN**	enables the B2 receive interrupts
31		**(reserved)**	

(see table 23 and 24 for bitmap explanation)

R_ST_D_IRQ_EN	(read / write)	0x000C0018

Interrupt enable register for S/T-HDLC D-channel
'0' = disable interrupt
'1' = enable interrupt

Bits	Reset Value	Name	Description
5..0	0	**V_D_TX_IRQ_EN**	enables the D-channel transmit interrupts
15..6	0	**V_D_RX_IRQ_EN**	enables the D-channel receive interrupts

(see table 23 (page 76) and 25 (page 77) for bitmap explanation)

R_ST_B1_TX_FIFO	(read / write)	0x000C001C

Write address of B1-channel transmitter FIFO (Organization: 16 x 32 bit = 64 bytes)

In transparent mode 4 bytes of data are written simultaneously. In HDLC mode 3 bytes of data and 1 control byte (byte 3) are written simultaneously.

Bits	Reset Value	Name	Description
7..0	0x00	**V_B1_TX_FIFO_BYTE0**	B1-channel transmit data byte 0
15..8	0	**V_B1_TX_FIFO_BYTE1**	B1-channel transmit data byt 1
23..16	0	**V_B1_TX_FIFO_BYTE2**	B1-channel transmit data byte 2
31..24	0x00	**V_B1_TX_FIFO_BYTE3**	in transp. mode: B1-channel transmit data byte 3 in HDLC mode: HDLC control byte

R_ST_B2_TX_FIFO	(read / write)	0x000C0020

Write address of B2-channel transmitter FIFO (Organization: 16 x 32 bit = 64 bytes)

In transparent mode 4 bytes of data are written simultaneously. In HDLC mode 3 bytes of data and 1 control byte (byte 3) are written simultaneously.

Bits	Reset Value	Name	Description
7..0	0	**V_B2_TX_FIFO_BYTE0**	B2-channel transmit data byte 0
15..8	0x00	**V_B2_TX_FIFO_BYTE1**	B2-channel transmit data byte 1
23..16	0x00	**V_B2_TX_FIFO_BYTE2**	B2-channel transmit data byte 2
31..24	0x00	**V_B2_TX_FIFO_BYTE3**	in transp. mode: B2-channel transmit data byte 3 in HDLC mode: HDLC control byte

R_ST_D_TX_FIFO	(read / write)	0x000C0024

Write address of D-channel transmitter FIFO (Organization: 16 x 32 bit = 64 bytes)

In transparent mode 4 bytes of data are written simultaneously. In HDLC mode 3 bytes of data and 1 control byte (byte 3) are written simultaneously.

Bits	Reset Value	Name	Description
7..0	0	**V_D_TX_FIFO_BYTE0**	D-channel transmit data byte 0
15..8	0	**V_D_TX_FIFO_BYTE1**	D-channel transmit data byte 1
23..16	0x00	**V_D_TX_FIFO_BYTE2**	D-channel transmit data byte 2
31..24	0x00	**V_D_TX_FIFO_BYTE3**	in transp. mode: D-channel transmit data byte 3 in HDLC mode: HDLC control byte

R_ST_B1_RX_FIFO	(read / write)	0x000C0028

Read address of B1-channel receiver FIFO (Organization: 16 x 32 bit = 64 bytes)

In transparent mode 4 bytes of data are read simultaneously. In HDLC mode 3 bytes of data and 1 control byte (byte 3) are read simultaneously.

Bits	Reset Value	Name	Description
7..0	0	**V_B1_RX_FIFO_BYTE0**	B1-channel receiver data byte 0
15..8	0	**V_B1_RX_FIFO_BYTE1**	B1-channel receiver data byte 1
23..16	0	**V_B1_RX_FIFO_BYTE2**	B1-channel receiver data byte 2
31..24	0x00	**V_B1_RX_FIFO_BYTE3**	in transp. mode: B1-channel receive data byte 3 in HDLC mode: HDLC control byte

R_ST_B2_RX_FIFO	(read / write)	0x000C0030

Read address of B2-channel receiver FIFO (Organization: 16 x 32 bit = 64 bytes)

In transparent mode 4 bytes of data are read simultaneously. In HDLC mode 3 bytes of data and 1 control byte (byte 3) are read simultaneously.

Bits	Reset Value	Name	Description
7..0	0	**V_B2_RX_FIFO_BYTE0**	B2-channel receiver data byte 0
15..8	0	**V_B2_RX_FIFO_BYTE1**	B2-channel receiver data byte 1
23..16	0	**V_B2_RX_FIFO_BYTE2**	B2-channel receiver data byte 2
31..24	0	**V_B2_RX_FIFO_BYTE3**	in transp. mode: B2-channel receive data byte 3 in HDLC mode: HDLC control byte

R_ST_D_RX_FIFO	(read / write)	0x000C0034

Read address of D-channel receiver FIFO (Organization: 16 x 32 bit = 64 bytes)

In transparent mode 4 bytes of data are read simultaneously. In HDLC mode 3 bytes of data and 1 control byte (byte 3) are read simultaneously.

Bits	Reset Value	Name	Description
7..0	0x00	**V_D_RX_FIFO_BYTE0**	D-channel receiver data byte 0
15..8	0x00	**V_D_RX_FIFO_BYTE1**	D-channel receiver data byte 1
23..16	0x00	**V_D_RX_FIFO_BYTE2**	D-channel receiver data byte 2
31..24	0x00	**V_D_RX_FIFO_BYTE3**	in transp. mode: D-channel receive data byte 3 in HDLC mode: HDLC control byte

R_ST_B1_CRC	(read / write)	0x000C0038

Register for soft CRC of B1-channel

If selected, the CRC has to be calculated by the software and stored into this register before the start of the HDLC frame.

Bits	Reset Value	Name	Description
15..0	0x0000	**V_ST_B1_CRC**	value of the CRC for the B1-channel

R_ST_B2_CRC	(read / write)	0x000C003C

Register for soft CRC of B2-channel

If selected, the CRC has to be calculated by the software and stored into this register before the start of the HDLC frame.

Bits	Reset Value	Name	Description
15..0	0x0000	**V_ST_B2_CRC**	value of the CRC for the B2-channel

R_ST_D_CRC	(read / write)	0x000C0040

Register for soft CRC of D-channel

If selected, the CRC has to be calculated by the software and stored into this register before the start of the HDLC frame.

Bits	Reset Value	Name	Description
15..0	0x0000	**V_ST_D_CRC**	value of the CRC for the D-channel

R_ST_CTRL	(read / write)	0x000C0044

FIFO control register

Bits	Reset Value	Name	Description
0	1	**V_B1_TX_FIFO_STOP**	stops the B1-channel transmit FIFO '0' = run '1' = stop If the FIFO is active but stopped, 0xFF will be transmitted.
1	1	**V_B2_TX_FIFO_STOP**	stops the B2-channel transmit FIFO '0' = run '1' = stop If the FIFO is active but stopped, 0xFF will be transmitted.
2	1	**V_B2_TX_FIFO_STOP**	stops the B2-channel transmit FIFO '0' = run '1' = stop If the FIFO is active but stopped, 0xFF will be transmitted.
3	1	**V_D_TX_FIFO_STOP**	stops the D-channel transmit FIFO '0' = run '1' = stop If the FIFO is active but stopped, 0xFF will be transmitted.
4	1	**V_B1_TX_FIFO_RDY**	resets the B1-channel transmit FIFO '0' = reset '1' = operation
5	1	**V_B2_TX_FIFO_RDY**	resets the B2-channel transmit FIFO '0' = reset '1' = operation
6	1	**V_D_TX_FIFO_RDY**	resets the D-channel transmit FIFO '0' = reset '1' = operation
7	1	**V_ST_RDY**	resets the S/T module '0' = reset '1' = operation
8		**(reserved)**	

R_ST_B12_STATUS	(read only)	0x000C0048

Status register for S/T-HDLC B1- and B2-channels
Represents the current status of the S/T-HDLC module

Bits	Reset Value	Name	Description
5..0	0x00	**V_B1_TX_STATUS**	indicates the B1-channel transmit FIFO status
14..6	0x000	**V_B1_RX_STATUS**	indicates the B1-channel transmit FIFO status
15		**(reserved)**	
21..16	0	**V_B2_TX_STATUS**	indicates the B2-channel transmit FIFO status
30..22	0	**V_B2_RX_STATUS**	indicates the B2-channel receive FIFO status
31		**(reserved)**	

(see table 23 (page 76) and 24 (page 76) for bitmap explanation)

R_ST_D_STATUS	(read only)	0x000C004C

Status register for S/T-HDLC D-channel
Represents the current status of the S/T-HDLC module

Bits	Reset Value	Name	Description
5..0	0x00	**V_D_TX_STATUS**	indicates the D-channel transmit FIFO status
15..6	0x000	**V_D_RX_STATUS**	indicates the D-channel transmit FIFO status

(see table 23 (page 76) and 25 (page 77) for bitmap explanation)

R_ST_RD_STATES	(read only)	0x000C00C0

Status register for S/T module state machine

Bits	Reset Value	Name	Description
3..0	0	**V_ST_STATE**	binary value of actual state (NT: Gx, TE: Fx)
4	0	**V_FR_SYNC**	frame synchronization '0' = not synchronized '1' = synchronized
5	0	**V_T2_EXP**	'1' = timer T2 expired (NT mode only)
6	0	**V_INFO0**	'1' = receiving INFO0
7	0	**V_G2_G3**	'0' = no operation '1' = allows transition from G2 to G3 in NT mode This bit is automatically cleared after the transition and has no function in TE mode.

Note: For bit 5 details see table 26 on page 92

R_ST_WR_STATES	(write only)	0x000C00C0

Status register for S/T module state machine

Bits	Reset Value	Name	Description
3..0	0	V_ST_SET_STATE	binary value of the new state (NT: Gx, TE: Fx) (bit 4 must also be set to load the state)
4	0	V_ST_LD_STATE	'1' loads the prepared state (bit 3 . . . 0) and stops the state machine. This bit needs to be set for a minimum period of 5.21 μs and must be cleared by software. '0' enables the state machine (bits 3 . . . 0 are ignored). After writing an invalid state, the state machine goes to deactivated state (G1, F2).
6..5	0	V_ST_ACT	'00' = no operation '01' = no operation '10' = start deactivation '11' = start activation These bits are automatically cleared after activation/deactivation.
7	0	V_SET_G2_G3	'0' no operation '1' in NT mode allows transition from G2 to G3. This bit is automatically cleared after the transition and has no function in TE mode.

Important !

The state machine is stuck to '0' after a reset. Writing a '0' to bit 4 restarts the state machine. In this state the HFC-S active sends no signal on the S/T line and it is not possible to activate it by incoming INFOx.

NT mode:

The NT state machine does not change automatically from G2 to G3 if the TE side sends INFO3 frames. This transition must be activated each time by setting bit 7 of the R_ST_WR_STATES register or by setting bit 0 of the R_ST_CTRL2 register.

Fix the NT state machine to state G3 when activated (by writing 13h into this register). This prevents deactivation of NT mode S/T interface due to sporadically errors on NT input data.

R_ST_CTRL1	(write only)	0x000C00C4

1st control register of the S/T module

Bits	Reset Value	Name	Description
0	0	**V_B1_EN**	'0' = B1 send data disabled (permanent 1 sent in activated states) '1' = B1 send data enabled
1	0	**V_B2_EN**	'0' = B2 send data disabled (permanent 1 sent in activated states) '1' = B2 send data enabled
2	0	**V_ST_MODE**	S/T interface mode '0' = TE mode '1' = NT mode
3	0	**V_D_PRIO**	D-channel priority '0' = high priority 8/9 '1' = low priority 10/11
4	0	**V_SQ_EN**	S/Q bit transmission '0' = S/Q bit disabled '1' = S/Q bit and multiframe enabled
5	0	**V_96KHZ**	'0' = normal operation '1' = send 96 kHz transmit test signal (alternating zeros)
6	0	**V_TX_LO**	TX2_LO and TX1_LO line setup This bit must be configured depending on the used S/T module and circuitry to match the 400Ω pulse mask test. '0' = capacitive line mode '1' = non capacitive line mode
7	0	**V_ST_STOP**	Power down '0' = power up, oscillator active '1' = power down, oscillator stopped **Note:** This bit is not cleared by a soft reset.

R_ST_CTRL2	(write only)	0x000C00C8

2nd control register of the S/T module

Bits	Reset Value	Name	Description
0	0	**V_G2_G3_EN**	force automatic transition from G2 to G3 without setting bit V_SET_G2_G3 of the R_ST_WR_STATES
1	0	**(reserved)**	must be '0'
2	0	**V_D_HI**	D reset '0' = normal operation '1' = D bits are forced to '1'
3	0	**V_E_IGNO**	D_U enable '0' = normal operation '1' = D-channel always sends enabled regardless of the received E bit
4	0	**V_E_LO**	force E to '0' (NT mode) '0' = normal operation '1' = E bit is forced to '0'
6..5	0	**(reserved)**	must be '00'
7	0	**V_B12_SWAP**	'0' = normal operation '1' = swap B1- and B2-channel in the S/T interface

R_ST_CTRL3	(write only)	0x000C00CC

3rd control register for the S/T module

Bits	Reset Value	Name	Description
0	0	**V_B1_RX_HI**	B1-channel receive enable '0' = B1 receive bits are forced to '1' '1' = normal operation
1	0	**V_B2_RX_HI**	B2-channel receive enable '0' = B2 receive bits are forced to '1' '1' = normal operation
7..2		**(reserved)**	

R_ST_SQ_MF	(read / write, read, write)	0x000C00D0

S/Q multiframe register for S/T module

Bits	Reset Value	Name	Description
3..0	0	**V_ST_SQ**	TE mode: S bits (bit 3 = S1, . . . , bit 0 = S4) NT mode: Q bits (bit 3 = Q1, bit 0 = Q4)
4	0	**V_MF_RX_RDY**	'1' a complete S or Q multiframe has been received Reading this register clears this bit.
6..5	0	**(reserved)**	
7	0	**V_MF_TX_RDY**	'1' ready to send a new S or Q multiframe Writing to this register clears this bit.

R_ST_CLK_CTRL	(write only)	0x000C00DC

Clock control register for S/T module

The register is not initialized with a '0' after reset. It should be initialized as follows before activating the TE/NT state machine:

TE mode: 0x0D . . . 0x0F

NT mode: 0x6C

Bits	Reset Value	Name	Description
3..0		**V_ST_CLK_DELAY**	TE: 4 bit delay value to adjust the 2 bit time between receive and transmit direction. The delay of the external S/T interface circuit can be compensated. The lower the value the smaller the delay between receive and transmit direction NT: Data sample point. The lower the value the earlier the input data is sampled. The steps are 163 ns.
6..4		**V_ST_SAMPLE**	NT mode only early edge input data shaping Low pass characteristic of extended bus configurations can be compensated. The lower the value the earlier input data pulse is sampled. No compensation means a value of 6 ('110'). Step size is 163 ns.
7		**(reserved)**	

R_ST_B1_RX	(read only)	0x000C00F0

Receive register for the B1-channel data. This register is updated all 125 μs (FSC pulse) by hardware.

Bits	Reset Value	Name	Description
7..0	0xFF	**V_ST_B1_RX**	B1-channel data

R_ST_B1_TX	(write only)	0x000C00F4

Transmit register for the B1-channel data. This register is automatically updated all 125 μs (FSC pulse) with data from the FIFO or from the switching unit or can be updated by the CPU if these data streams are not configured.

Bits	Reset Value	Name	Description
7..0	0x00	**V_ST_B1_TX**	B1-channel data

R_ST_B2_RX	(read only)	0x000C00F8

Receive register for the B2-channel data. This register is updated all 125 μs (FSC pulse) by hardware.

Bits	Reset Value	Name	Description
7..0	0xFF	**V_ST_B2_RX**	B2-channel data

R_ST_B2_TX	(write only)	0x000C00FC

Transmit register for the B2-channel data. This register is automatically updated all 125 μs (FSC pulse) with data from the FIFO or from the switching unit or can be updated by the CPU if these data streams are not configured.

Bits	Reset Value	Name	Description
7..0	0x00	**V_ST_B2_TX**	B2-channel data

R_ST_D_RX	(read only)	0x000C0100

Receive register for the D-channel data. This register is updated all 125 μs (FSC pulse) by hardware.

Bits	Reset Value	Name	Description
5..0		**(reserved)**	
7..6	3	**V_ST_D_RX**	D-channel data

R_ST_D_TX	(write only)	0x000C0104

Transmit register for the D-channel data. This register is automatically updated all 125 μs (FSC pulse) with data from the FIFO or from the switching unit or can be updated by the CPU if these data streams are not configured.

Bits	Reset Value	Name	Description
5..0		**(reserved)**	
7..6	0	**V_ST_D_TX**	D-channel data

R_ST_E_RX	(read only)	0x000C0108

Receive register for the E-channel data. This register is updated all 125 μs (FSC pulse) by hardware.

Bits	Reset Value	Name	Description
5..0		**(reserved)**	
7..6	3	**M_ST_E_RX**	E-channel data

4.2.3 State matrices for NT and TE

S/T interface activation / deactivation layer 1 for finite state matrix for NT

Table 26: *Activation / deactivation layer 1 for finite state matrix for NT*

State name:	Reset	Deactivate	Pending activation	Active	Pending deactivation
State number:	**G 0**	**G 1**	**G 2**	**G 3**	**G 4**
INFO sent:	**INFO 0**	**INFO 0**	**INFO 2**	**INFO 4**	**INFO 0**
Event:					
State machine release (Note 3)	G 2	|	|	|	|
Activate request	G 2 (Note 1)	G 2 (Note 1)	|	|	G 2 (Note 1)
Deactivate request	—	|	Start timer T2 G 4	Start timer T2 G 4	|
Expiry T2 (Note 2)	—	—	—	—	G1
Receiving INFO 0	—	—	—	G 2	G 1
Receiving INFO 1	—	G 2 (Note 1)	—	/	—
Receiving INFO 3	—	/	G 3 (Note 1, 4)	—	—
Lost framing	—	/	/	G2	—

Legend:

— No state change

/ Impossible by the definition of peer-to-peer physical layer procedures or system internal reasons

| Impossible by the definition of the physical layer service

Notes:

Note 1: Timer 1 (T1) is not implemented in the HFC-S active and must be implemented in software.

Note 2: Timer 2 (T2) prevents unintentional reactivation. Its value is 32 ms ($256 \cdot 125\,\mu s$). This implies that a TE has to recognize INFO 0 and to react on it within this time.

Note 3: After reset the state machine is fixed to G0.

Note 4: Bit V_SET_G2_G3 of the R_ST_WR_STATES register must be set to allow this transition.

Activation / deactivation layer 1 for finite state matrx for TE

Table 27: *Activation / deactivation layer 1 for finite state matrix for TE*

State name:	Reset	Sensing	Deactivated	Awaiting signal	Identifying input	Synchronized	Activated	Lost framing
State number:	**F 0**	**F 2**	**F 3**	**F 4**	**F 5**	**F 6**	**F 7**	**F 8**
INFO sent:	**INFO 0**	**INFO 0**	**INFO 0**	**INFO 1**	**INFO 0**	**INFO 3**	**INFO 3**	**INFO 0**
Event:								
State machine release (Note 1)	F 2	/	/	/	/	/	/	/
Activate request,								
receiving any signal	—	|	F 5	|	|	—	|	—
receiving INFO 0	—	|	F 4	|	|	—	|	—
Expiry T3 (Note 5)	—	/	—	F 3	F 3	F 3	—	—
Receiving INFO 0	—	F 3	—	—	—	F 3	F 3	F 3
Receiving any signal (Note 2)	—	—	—	F 5	—	/	/	—
Receiving INFO 2 (Note 3)	—	F 6	F 6	F 6	F 6	—	F 6	F 6
Receiving INFO 4 (Note 3)	—	F 7	F 7	F 7	F 7	F 7	—	F 7
Lost framing (Note 4)	—	/	/	/	/	F 8	F 8	—

Legend:

— No state change

/ Impossible situation

| Impossible by the definition of the layer 1 service

Notes:

Note 1: After reset the state machine is fixed to F 0.

Note 2: This event reflects the case where a signal is received and the TE has not (yet) determined wether it is INFO 2 or INFO 4.

Note 3: Bit- and frame-synchronisation achieved.

Note 4: Loss of Bit- or frame-synchronisation.

Note 5: Timer 3 (T3) is not implemented in the HFC-S active and must be implemented in software.

4.2.4 Binary organisation of the frame

The frame structures on the S/T interface are different for each direction of transmission. Both structures are illustrated in figure 22.

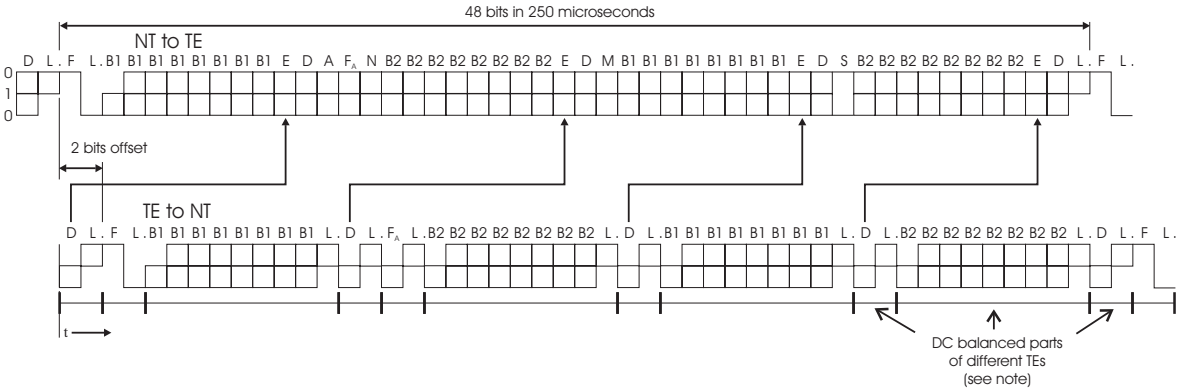


Figure 22: *Frame structure at reference point S and T*

Legend:

Code	Explanation	Code	Explanation
F	Framing bit	N	Bit set to a binary value $N = \overline{F}_A$ (NT to TE)
L	D.C. balancing bit	B1	Bit within B1-channel
D	D-channel bit	B2	Bit within B2-channel
E	D-echo-channel bit	A	Bit used for activation
F_A	Auxiliary framing bit	S	S-channel bit
M	Multiframing bit		

Note !

Lines demarcate those parts of the frame that are independently d.c.-balanced.

The F_A bit in the direction TE to NT is used as Q bit in every fifth frame if S/Q bit transmission is enabled (see R_ST_CTRL1 register).

The nominal 2 bit offset is as seen from the TE. The offset can be adjusted with the R_ST_CLK_CTRL register in TE mode. The corresponding offset at the NT may be greater due to delay in the interface cable and varies by configuration.

HDLC B-channel data start with the LSB, PCM B-channel data start with the MSB.

4.2.5 S/T interface circuitry

In order to comply to the physical requirements of ITU-T recommendation I.430 and considering the national requirements concerning overvoltage protection and electromagnetic compatibility (EMC), the HFC-S active needs some additional circuitry, which are shown in appendix D (see page 162).

A list of suitable S/T modules is given in the tables 28 and 29. Furtheron, an actual list of S/T modules is always available on the web site `www.CologneChip.com`.

Table 28: *S/T module part numbers and manufacturers (part 1)*

S/T module part number	Manufacturer
APC 56624-1 APC 40495S (SMD) S-Hybrid modules with receiver and transmitter circuitry included: APC 5568-3V APC 5568-5V APC 5568DS-3V APC 5568DS-5V	**Advanced Power Components** *United Kingdom* Phone: +44 1634-290588 Fax: +44 1634-290591 URL: http://www.apcisdn.com
FE 8131-55Z	**FEE GmbH** *Singapore* Phone: +65 741-5277 Fax: +65 741-3013 *Bangkok* Phone: +662 718-0726-30 Fax: +662 718-0712 *Germany* Phone: +49 6106-82980 Fax: +49 6106-829898
transformers: PE-64995 PE-64999 PE-65795 (SMD) PE-65799 (SMD) PE-68995 PE-68999 T5006 (SMD) T5007 (SMD) S_0-modules: T5012 T5034 T5038	**Pulse Engineering, Inc.** *United States* Phone: +1-619-674-8100 Fax: +1-619-674-8262 URL: http://www.pulseeng.com
transformers: SM TC-9001 SM ST-9002 SM ST-16311F S_0-modules: SM TC-16311 SM TC-16311A	**Sun Myung** *Korea* Phone: +82-348-943-8525 Fax: +82-348-943-8527 URL: http://www.sunmyung.com

Table 29: *S/T module part numbers and manufacturers (part 2)*

S/T module part number	Manufacturer
transformers UT21023 S_0-modules: UT 20795 (SMD) UT 21624 UT 28624 A	**UMEC GmbH** *Germany* Phone: +49 7131-7617-0 Fax: +49 7131-7617-20 *Taiwan* Phone: +886-4-359-009-6 Fax: +886-4-359-012-9 *United States* Phone: +1-310-326-707-2 Fax: +1-310-326-705-8 URL: http://www.umec.de
all devices T 6040… transformers: …3-L4021-X066 …3-L4025-X095 …3-L5024-X028 …3-L4096-X005 …3-L5032-X040 S_0-modules: …7-L5026-X010 (SMD) …7-L5051-X014 …7-M5051-X032 …7-L5052-X102 (SMD) …7-M5052-X110 …7-M5052-X114	**VAC GmbH** *Germany* Phone: +49 6181/ 38-0 Fax: +49 6181/ 38-2645 URL: http://www.vacuumschmelze.de
transformers: ST5069 S_0-modules: PT5135 ST5201 ST5202	**Valor Electronics, Inc.** *Asia* Phone: +852 2333-0127 Fax: +852 2363-6206 *North America* Phone: +1 800 31VALOR Fax: +1 619 537-2525 *Europe* Phone: +44 1727-824-875 Fax: +44 1727-824-898 URL: http://www.valorinc.com
543 76 009 00 503 740 010 0 (SMD)	**Vogt electronic AG** *Germany* Phone: +49 8591/ 17-0 Fax: +49 8591/ 17-240 URL: http://www.vogt-electronic.com

4.3 PCM highway module

Table 30: *Overview of the* HFC-S active *PCM highway pins (primary function for pins marked with *)*

Number	Name	Description	Number	Name	Description
93	SDI0	serial data input for PCM highway 1	101 *	SDI1	serial data input for PCM highway 2
94	SDO0	serial data output for PCM highway 1	102 *	SDO1	serial data output for PCM highway 2
95	BCLK0	bit clock for PCM highway 1	103 *	BCLK1	bit clock for PCM highway 2
96	FSC0	frame sync signal for PCM highway 1	104 *	FSC1	frame sync signal for PCM highway 2
97	PFS3	peripheral frame sync signal	105 *	SDI2	serial data input for PCM highway 3
98	PFS2	peripheral frame sync signal	106 *	SDO2	serial data output for PCM highway 3
99	PFS1	peripheral frame sync signal	109 *	BCLK2	bit clock for PCM highway 2
100	PFS0	peripheral frame sync signal	110 *	FSC2	frame sync signal for PCM highway 3

Table 31: *Overview of the* HFC-S active *PCM highway registers (* : The bit V_CODEC_ST is part of the* Switching Unit*, see section 4.4)*

Address	Name	Page	Address	Name	Page
0x000B0000	R_HW1_TX_NEXT	103	0x000B01A0	R_HW3_RX_CUR	107
0x000B0020	R_HW1_RX_LAST	103	0x000B01F4	R_PFS0_CFG	108
0x000B0040	R_HW2_TX_NEXT	104	0x000B01F6	R_PFS1_CFG	108
0x000B0060	R_HW2_RX_LAST	104	0x000B01F8	R_PFS2_CFG	109
0x000B0080	R_HW3_TX_NEXT	104	0x000B01FA	R_PFS3_CFG	109
0x000B00A0	R_HW3_RX_LAST	105	0x000B0200	R_PCM_CFG *	110
0x000B0100	R_HW1_TX_CUR	105	0x000B0204	R_HW1_CTRL	111
0x000B0120	R_HW1_RX_CUR	105	0x000B0208	R_HW2_CTRL	113
0x000B0140	R_HW2_TX_CUR	106	0x000B020C	R_HW3_CTRL	115
0x000B0160	R_HW2_RX_CUR	106	0x000B0210	R_HW_SL_CNT	116
0x000B0180	R_HW3_TX_CUR	106			

4.3.1 Overview

The PCM highway module provides three PCM highway interfaces with a data rate of up to 2.048 Mbit/s each. This allows the connection with external PCM or IOM-2 compatible devices (see table32 for name mapping) or to cascade the HFC-S active for advanced applications. The data stream of the

Table 32: *Name mapping between* HFC-S active *PCM interface pins and the corresponding IOM-2 abbreviations (DCL = data clock, FSC = frame synchronization clock, DD = data downstream, DU = data upstream)*

HFC-S active name	IOM-2 name	Description
BCLKx	DCL	Bus clock output
FSCx	FSC	Frame clock output (always 8 kHz)
SDOx	DD	Data output
SDIx	DU	Data input

PCM highway interface is divided into 32 time slots (channels). Each time slot can receive and transmit 1 byte per FSC pulse (125 μs). Only with the maximum PCM data rate it is possible to use all 32 time slots. With lower data rates, the number of available time slots is reduces accordingly.

The three interfaces receive the bit and frame clock signals from the internal FSC-PLL of the HFC-S active, so that the PCM highways always work synchronously to the S/T module and the CODEC module. The following parameters of the PCM highways are programmable to allow a flexible operation:

- Bit rate 256 kbit/s ... 2.048 Mbit/s
- Used time slots for transmission
- Single or double clocking
- Phase position and pulse length of the frame synchronization signals PFS0 ... PFS3
- Open drain output for 5 V compatibility[6]
- The characteristic of turn around cycles

The three PCM highways can be controlled completely independent, so every parameter can be set for each PCM highway individually.

The standard mode of the PCM highway interface is usually the master mode. In master mode, bit clocks (pins BCLK0, BCLK1 and BCLK2) and the frame synchronization signals (FSC0, FSC1 and FSC2) are driven by the HFC-S active. In slave mode the FSC and BCLK signals are driven by an external device (e.g. an other HFC-S active in master mode). Due to the possibility to configure each PCM highway individually to master or slave mode, it is possible to build cascaded networks with HFC-S active chips. By the means of high level protocols it is possible to build ISDN networks with arbitrary complexity and topology.

If a highway is not used its SDI pin must have a defined potential. Highway 3 shares its pins with the internal UART, so only one of these interfaces can be used at a time.

4.3.2 Switching buffer mechanism

The data transfer between the PCM highways and the ARM7^{TM} CPU is carried out via a switching buffer mechanism. Figure 23 illustrates the scheme of the PCM data flow.

[6] ... via external pull-up resistor to a 5 V source. In this case the pin SDO0 (resp. SDO1, SDO2) have to be switched to `high-Z`. This is achieved with the register R_HW1_TS_EN (resp. R_HW2_TS_EN, R_HW3_TS_EN) for each time slot independently. The input ports of the PCM highways are 5 V tolerant.

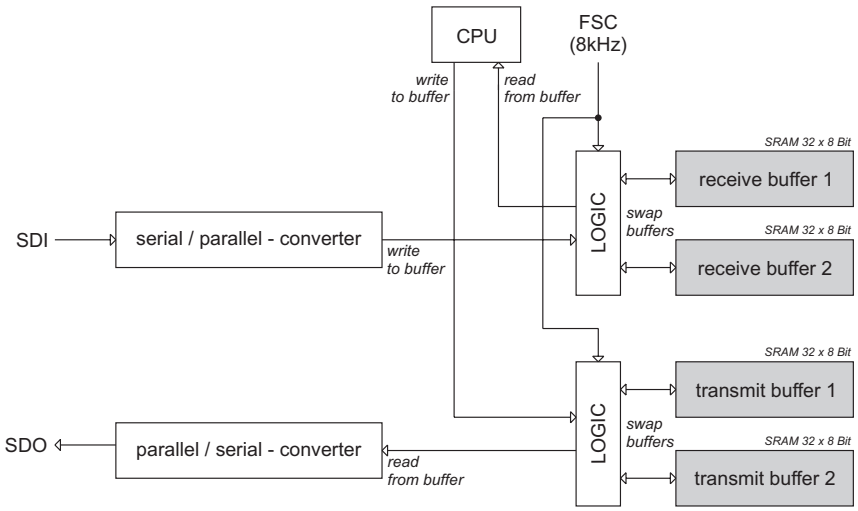


Figure 23: *The scheme of the PCM highway interface*

The receive and transmit buffers exist in duplicate each. So the ARM7^{TM} CPU can access one buffer pair while the PCM interface operates on the other buffer pair at the same time without the risk of collisions. As an instance of the PCM highway 1, at every FSC pulse (8 kHz) the HFC-S active exchanges the R_HW1_TX_CUR buffer with R_HW1_TX_NEXT buffer (resp. R_HW1_RX_CUR with R_HW1_RX_LAST for receive direction) automatically. So the ARM7^{TM} CPU can always write to the register R_HW1_TX_NEXT and read from R_HW1_RX_LAST while the PCM highway interface writes to R_HW1_TX_CUR and reads from R_HW1_RX_CUR at the same time.

If required, the switching buffer functionality can be disabled by software. In this case the software must ensure a collision-free data handling as both, the PCM highway and the ARM7^{TM} CPU, write to R_HW1_TX_CUR and read from R_HW1_RX_CUR.

The PCM highway transmit buffer is implemented as a single port RAM. Therefore three waitstates must be programmed in the corresponding waitstates register at least, to assure a proper data exchange between the CPU and the PCM highway interface. The waitstates must be adjusted to the clock frequency ratio of the CPU and the PCM highway.

4.3.3 Time slot configuration

The number of available time slots depends on the selected PCM data rate. This can be cofigured for each PCM interface independentliy, e.g. for the PCM highway 1 with the bitmap V_HW1_BR of the register R_HW1_CTRL. Four data rates are available and the number of time slots is

$$n = \frac{[datarate]\, kBit/s}{64\,\mathrm{kBit/s}}\,.$$

After HFC-S active reset all time slots are switched off. The time slots can be activated by writing a '1' to the appropriate bit of the registers R_HW1_TS_EN, R_HW2_TS_EN or R_HW1_TS_EN.

Additionally, time slots can be assigned to the S/T interface and the CODECs. This functionallity is part of the HFC-S actives *Switching Unit* (see section 4.4).

4.3.4 Peripheral frame synchronization signals

The first PCM highway has four freely programmable peripheral frame synchronization outputs PFS0... PFS3. By the peripheral FSC signals each time slot can be selected for the external peripheral devices (e.g. additional external CODEC). The start position and the length of the PFSC is programmable (see R_PFS0_CFG... R_PFS3_CFG register description).

4.3.5 Enabling a PCM highway

Before a PCM highway can be used some initial settings must be done[7].

1. The system frequency in the interface control register R_PCM_CFG must be specified.
2. The PCM highway control register read / writeust be configured:
   - PCM data rate (bit V_HW1_BR)
   - Single- or double-bit clocking on BCLK0 output (bit V_HW1_DCLK)
   - Rising or falling edge of transmit and receive date (bits V_HW1_TX_EDGEand V_HW1_RX_EDGE)
   - Number of turn around cycles (bitmap V_BCLK0_WAIT), V_SD0_WAITmust be set to '1' if turn around cycles are greater than zero
   - Master or slave mode of the PCM highway (bit V_HW1_MASTER)
   - V_HW1_SDO0_EN = '1' if the PCM highway has to run with a permanently data stream on SDO0, with V_HW1_SDO0_EN = '0' certain time slots can be enabled or disabled independently. In this functional setting there are further configuration options, i.e. a *switching unit* allows to connect the HFC-S active CODESCs und S/T interface to arbitrary time slots (see section 4.4).
   - Switching buffer enable or disable (bit V_HW1_BUFF_OFF)
3. The switching buffers must be initialized, i.e. the registers R_HW1_TX_CUR and R_HW1_TX_NEXT should get meaningful contents.
4. For the PCM highway 1 only, the peripheral FSC registers R_PFS0_CFG ... R_PFS3_CFG have to be initialized.
5. The bit clock BCLK0 has to be enabled by setting V_HW1_BCLK_EN = '1'.
6. Finally, V_HW1_FSC0_EN = '1' enables the FSC0 clock signal. After this bit has been set the BCLK0 signal is started after the next FSC pulse.

Changes to the system frequency, data rate and bit clock take effect immediately and may not be done while the highway is enabled.

The figures 24 to 26 show some timing examples for different highway settings which are listed in table 33.

[7] This section describes the highway 1 settings and is valid for the other two PCM highways in the same way.

Table 33: *Configuration settings for the timing examples in figures 24 to 26*

Figure	V_HW1_DCLK	V_HW1_TX_EDGE	V_HW1_RX_EDGE	V_SD0_WAIT	V_BCLK0_WAIT
24	'0' (single)	'0' (rising edge)	'0' (falling edge)	'1' (enable)	'001' (1 wait cycle)
25	'0' (single)	'0' (rising edge)	'1' (rising edge)	'1' (enable)	'011' (3 wait cycles)
26	'1' (double)	'1' (falling edge)	'1' (rising edge)	'0' (disable)	'xxx' (no wait cycles)

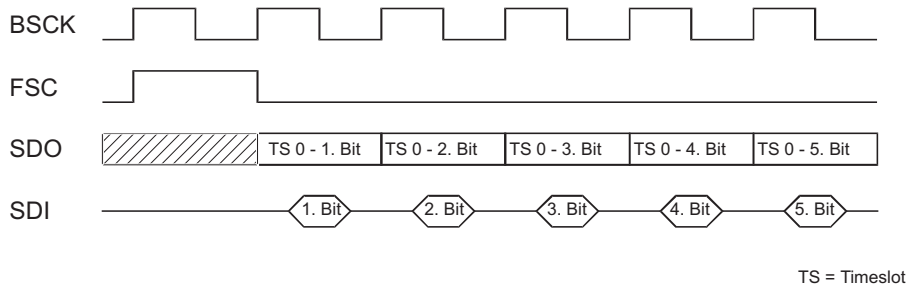


Figure 24: *PCM timing with the configuration shown in table 33 (1st line)*

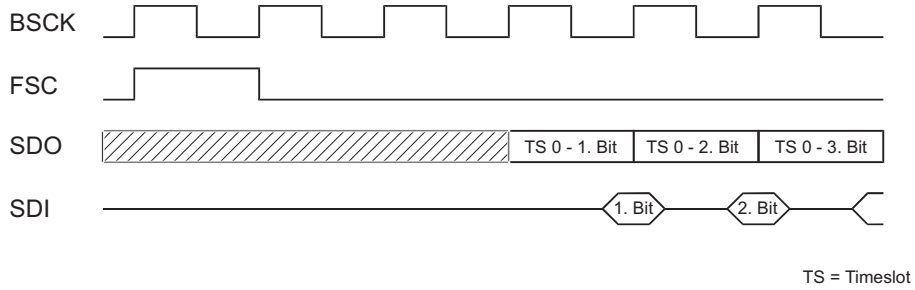


Figure 25: *PCM timing with the configuration shown in table 33 (2nd line)*

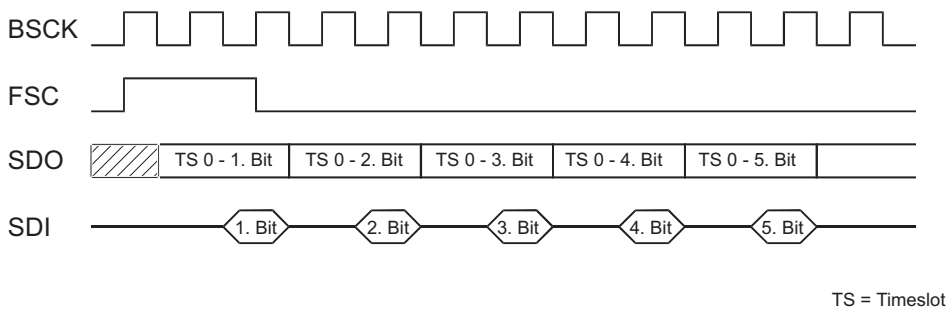


Figure 26: *PCM timing with the configuration shown in table 33 (3rd line)*

Note !

Exemplary for the PCM highway 1, the bit clock BCLK0 is required for the data transfer and must be enabled first. If both V_HW1_BCLK_EN and V_HW1_FSC0_EN of the register R_HW1_CTRL are set at the same time, the operation is as follows:

1. After the 1st FSC pulse the bit clock BCLK0 (... BCLK2) will be enabled.
2. After the 2nd FSC pulse also the data transfer on the highway will be enabled.

(This note is valid for the other two PCM highways in the same way.)

4.3.6 Disabling a PCM highway

To ensure a proper data transfer halt, the PCM highway should be disabled in two steps (described for PCM highway 1, also valid for the other two PCM highways):

1. Setting V_HW1_FSC0_EN = 0 the data transfer will be disabled after the next FSC pulse. The bit clock BCLK0 will remain available.
2. Setting V_HW1_BCLK_EN = 0 the bit clock BCLK0 will be stopped immediately.

If the first point is not executed before the second one, the data transmisson may be stopped right in the middle of a byte!

4.3.7 Register description

R_HW1_TX_NEXT	(read / write)	0x000B0000

Base address of the PCM Highway 1 next send register

The send buffer operates as a switching buffer. So the CPU can write to the next transmit buffer and the hardware can send the current data without any collisions. At every FSC pulse (8 kHz) the HFC-S active changes the R_HW1_TX_NEXT buffer to R_HW1_TX_CUR buffer automatically. If required, the switching buffer functionality can be disabled by software.

The PCM highway transmit buffer is implemented as a single port RAM. Therefore three waitstates must be programmed in the corresponding waitstates register at least, to assure a proper data exchange between the CPU and the PCM highway interface. The waitstates must be adjusted to the clock frequency ratio of the CPU and the PCM highway.

The transmit buffer RAM supports 8 / 16 / 32 bit access.

Bits	Reset Value	Name	Description
31..0	0	**V_HW1_TX_NEXT**	the access to address 0x000B0000 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW1_RX_LAST	(read / write)	0x000B0020

Base address of the PCM Highway 1 last receive register.

Bits	Reset Value	Name	Description
31..0	0	**V_HW1_RX_LAST**	the access to address 0x000B0020 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW2_TX_NEXT	(read / write)	0x000B0040

Base address of the PCM Highway 2 next send register

The send buffer operates as a switching buffer. So the CPU can write to the next transmit buffer and the hardware can send the current data without any collisions. At every FSC pulse (8 kHz) the HFC-S active changes the R_HW2_TX_NEXT buffer to R_HW2_TX_CUR buffer automatically. If required the switching buffer functionality can be disabled by software.

Bits	Reset Value	Name	Description
31..0	0	**V_HW2_TX_NEXT**	the access to address 0x000B0040 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW2_RX_LAST	(read / write)	0x000B0060

Base address of the PCM Highway 2 last receive register

Bits	Reset Value	Name	Description
31..0	0	**V_HW2_RX_LAST**	the access to address 0x000B0060 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW3_TX_NEXT	(read / write)	0x000B0080

Base address of the PCM Highway 3 next send register

The send buffer operates as a switching buffer. So the CPU can write to the next transmit buffer and the hardware can send the current data without any collisions. At every FSC pulse (8 kHz) the HFC-S active changes the R_HW3_TX_NEXT buffer to R_HW3_TX_CUR buffer automatically. If required the switching buffer functionality can be disabled by software.

Bits	Reset Value	Name	Description
31..0	0	**V_HW3_TX_NEXT**	the access to address 0x000B0080 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW3_RX_LAST	(read / write)	0x000B00A0

Base address of the PCM Highway 3 last receive register.

Bits	Reset Value	Name	Description
31..0	0	**V_HW3_RX_LAST**	the access to address 0x000B00A0 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW1_TX_CUR	(read / write)	0x000B0100

Base address of the PCM highway 1 current send register. The hardware prevent the collision automatically. By writing to this register the software has to take care of the consistence of the data. This register must be used, if the switching buffer is disabled.

Bits	Reset Value	Name	Description
31..0	0	**V_HW1_TX_CUR**	the access to address 0x000B0100 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW1_RX_CUR	(read / write)	0x000B0120

Base address of the PCM highway 1 current receive register. The hardware prevent the collision automatically. By writing to this register the software has to take care of the consistence of the data. This register must be used, if the switching buffer is disabled.

Bits	Reset Value	Name	Description
31..0	0	**V_HW1_RX_CUR**	the access to address 0x000B0120 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW2_TX_CUR	(read / write)	0x000B0140

Base address of the PCM highway 2 current send register. The hardware prevent the collision automatically. By writing to this register the software has to take care of the consistence of the data. This register must be used, if the switching buffer is disabled.

Bits	Reset Value	Name	Description
31..0	0	**V_HW2_TX_CUR**	the access to address 0x000B0140 $+n \cdot 4$ selects the time slots $TS[n \cdot 4] \ldots TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0 \ldots 7$)

R_HW2_RX_CUR	(read / write)	0x000B0160

Base address of the PCM highway 2 current receive register. The hardware prevent the collision automatically. By writing to this register the software has to take care of the consistence of the data. This register must be used, if the switching buffer is disabled.

Bits	Reset Value	Name	Description
31..0	0	**V_HW2_RX_CUR**	the access to address 0x000B0160 $+n \cdot 4$ selects the time slots $TS[n \cdot 4] \ldots TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0 \ldots 7$)

R_HW3_TX_CUR	(read / write)	0x000B0180

Base address of the PCM highway 3 current send register. The hardware prevent the collision automatically. By writing to this register the software has to take care of the consistence of the data. This register must be used, if the switching buffer is disabled.

Bits	Reset Value	Name	Description
31..0	0	**V_HW3_TX_CUR**	the access to address 0x000B0180 $+n \cdot 4$ selects the time slots $TS[n \cdot 4] \ldots TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0 \ldots 7$)

R_HW3_RX_CUR	(read / write)	0x000B01A0

Base address of the PCM highway 3 current receive register. The hardware prevent the collision automatically. By writing to this register the software has to take care of the consistence of the data. This register must be used, if the switching buffer is disabled.

Bits	Reset Value	Name	Description
31..0	0	**V_HW3_RX_CUR**	the access to address 0x000B01A0 $+n \cdot 4$ selects the time slots $TS[n \cdot 4]$ ... $TS[n \cdot 4 + 3]$ in byte 0 ... byte 3 ($n = 0$ ... 7)

R_HW1_TS_EN	(read / write)	0x000B01E8

Time slot enable register

The bit number is equal to the timslot number.
'0' = port SDO is high-Z
'1' = time slot is active

Bits	Reset Value	Name	Description
31..0	0x00000000	**V_HW1_TS_EN**	enable register of highway 1 time slots for transmission

R_HW2_TS_EN	(read / write)	0x000B01EC

Time slot enable register

The bit number is equal to the timslot number.
'0' = port SDO is high-Z
'1' = time slot is active

Bits	Reset Value	Name	Description
31..0	0x00000000	**V_HW2_TS_EN**	enable register of highway 2 time slots for transmission

R_HW3_TS_EN	(read / write)	0x000B01F0

Time slot enable register

The bit number is equal to the timslot number.
'0' = port SDO is high-Z
'1' = time slot is active

Bits	Reset Value	Name	Description
31..0	0x00000000	**V_HW3_TS_EN**	enable register of highway 3 time slots for transmission

R_PFS0_CFG	(read / write)	0x000B01F4

Control register for the frame synchronization signal on PFS0 port

Bits	Reset Value	Name	Description
8..0	0	**V_PFS0_START**	defines the start position of the PFS0 signal Note: The value refers to the number of clock cycles of the BCLK signal.
15..9	0	**V_PFS0_LEN**	defines the length of the PFS0 signal Note: The value refers to the number of clock cycles of the BCLK signal.

R_PFS1_CFG	(read / write)	0x000B01F6

Control register for the frame synchronization signal on PFS1 port

Bits	Reset Value	Name	Description
8..0	0	**V_PFS1_START**	defines the start position of the PFS1 signal Note: The value refers to the number of clock cycles of the BCLK signal.
15..9	0	**V_PFS1_LEN**	defines the length of the PFS1 signal Note: The value refers to the number of clock cycles of the BCLK signal.

R_PFS2_CFG	(read / write)	0x000B01F8

Control register for the frame synchronization signal on PFS2 port

Bits	Reset Value	Name	Description
8..0	0	**V_PFS2_START**	defines the start position of the PFS2 signal Note: The value refers to the number of clock cycles of the BCLK signal.
15..9	0	**V_PFS2_LEN**	defines the length of the PFS2 signal Note: The value refers to the number of clock cycles of the BCLK signal.

R_PFS3_CFG	(read / write)	0x000B01FA

Control register for the frame synchronization signal on PFS3 port

Bits	Reset Value	Name	Description
8..0	0	**V_PFS3_START**	defines the start position of the PF3 signal Note: The value refers to the number of clock cycles of the BCLK signal.
15..9	0	**V_PFS3_LEN**	defines the length of the PFS3 signal Note: The value refers to the number of clock cycles of the BCLK signal.

R_PCM_CFG	(read / write)	0x000B0200

Control register for PCM highway interface configuration.

Bits	Reset Value	Name	Description
2..0	0	**V_PCM_CLK**	defines the operating frequency of the PCM highway interfaces Coding: '000' 12.288 MHz '001' 24.576 MHz '010' 36.864 MHz '100' 61.440 MHz Note: The PCM highway interface gets the external (12.288 MHz) system clock always. At 12.288 MHz system frequency these bits should be left on zero.
3	0	**V_CODEC_ST**	enables the coupling between S/T and CODECs '0' = disable '1' = enable
7..4		**(reserved)**	

R_HW1_CTRL	(read / write)	0x000B0204

Control register for PCM Highway 1

Bits	Reset Value	Name	Description
0	0	**V_HW1_BCLK_EN**	enables the clock signal on BCLK0 (bit clock) port '0' =off '1' = on
1	0	**V_HW1_FSC0_EN**	enables the frame synchronization signal on FSC0 port '0' = off '1' = on
2	0	**V_HW1_EN**	enables the switching unit for the PCM Highway 1 '0' = off '1' = on
4..3	0	**V_HW1_BR**	defines the transmission rate on the PCM Highwa 1 Coding: '00' 2048 kBit/s '01' 768 kBit/s '10' 512 kBit/s '11' 256 kBit/s
5	0	**V_HW1_DCLK**	enables the double clocking mode on PCM Highway 1 '0' = single clock '1' = double clock
6	0	**V_HW1_TX_EDGE**	defines the output edge of sending data '0' = rising edge '1' = falling edge
7	0	**V_HW1_RX_EDGE**	defines the sampling edge of receiving data '0' = falling edge '1' = rising edge
8	0	**V_SD0_WAIT**	enables turn around cycles (wait cycles between FSC and first data bit) on SDO0 port '0' = disable '1' = enable
11..9	0	**V_BCLK0_WAIT**	defines the number of turn around cycles in BCCK0 clock units ('000' = 0 and so on)
12	0	**V_HW1_MASTER**	Defines the mode for PCM Highway 1. In slave mode the port FSC0 and BCLK0 are not driven by the HFC-S active (high-Z). '0' = slave '1' = master
13	0	**V_HW1_SDO0_EN**	Enables permanently the SDO0 Port of the PCM Highway 1. If enable the entry in the R_HW1_TS_EN has no influence. '0' = disable '1' = enable

Bits	Reset Value	Name	Description
14	0	**V_HW1_BUFF_OFF**	disables the switching buffer for the PCM Highway 1 '1' = off '0' = on
15		**(reserved)**	

R_HW2_CTRL	(read / write)	0x000B0208

Control register for PCM Highway 2

Bits	Reset Value	Name	Description
0	0	**V_HW2_BCLK_EN**	enables the clock signal on BCLK1 (bit clock) port '0' = off '1' = on
1	0	**V_HW2_FSC1_EN**	enables the frame synchronization signal on FSC1 port '0' = off '1' = on
2	0	**V_HW2_EN**	enables the switching unit for the PCM Highway 2 '0' = off '1' = on
4..3	0	**V_HW2_BR**	defines the transmission rate on the PCM Highway 2 Coding: '00' 2048 kBit/s '01' 768 kBit/s '10' 512 kBit/s '11' 256 kBit/s
5	0	**V_HW2_DCLK**	enables the double clocking mode on PCM Highway 2 '0' = single clock '1' = double clock
6	0	**V_HW2_TX_EDGE**	defines the output edge of sending data '0' = rising edge '1' = falling edge
7	0	**V_HW2_RX_EDGE**	defines the sampling edge of receiving data '0' = falling edge '1' = rising edge
8	0	**V_SD1_WAIT**	enables turn around cycles (wait cycles between FSC and first data bit) on SDO1 port '0' = disable '1' = enable
11..9	0	**V_BCLK1_WAIT**	defines the number of turn around cycles in BCCK1 clock units ('000' = 0 and so on)
12	0	**V_HW2_MASTER**	Defines the mode for PCM Highway 2. In slave mode the port FSC1 and BCK1 are not driven by the HFC-S active (high-Z). '0' = slave '1' = master
13	0	**V_HW2_SDO1_EN**	Enables permanently the SDO1 Port of the PCM Highway 2. If enabled, the entry in the R_HW2_TS_EN has no influence. '0' = disable '1' = enable

Bits	Reset Value	Name	Description
14	0	**V_HW2_BUFF_OFF**	disables the switching buffer for the PCM Highway 2 '1' = off '0' = on
15		**(reserved)**	

R_HW3_CTRL	(read / write)	0x000B020C
Control register for PCM Highway 3		

Bits	Reset Value	Name	Description
0	0	**V_HW3_BCLK_EN**	Enables the clock signal on BCLK2 (bit clock) port. '0' = off '1' = on
1	0	**V_HW3_FSC2_EN**	enables the frame synchronization signal on FSC2 port '0' = off '1' = on
2	0	**V_HW3_EN**	enables the switching unit for the PCM Highway 3 '0' = off '1' = on
4..3	0	**V_HW3_BR**	defines the transmission rate on the PCM Highway 3 Coding: '00' 2048 kBit/s '01' 768 kBit/s '10' 512 kBit/s '11' 256 kBit/s
5	0	**V_HW3_DCLK**	enables the double clocking mode on PCM Highway 3 '0' = single clock '1' = double clock
6	0	**V_HW3_TX_EDGE**	defines the output edge of sending data '0' = rising edge '1' = falling edge
7	0	**V_HW3_RX_EDGE**	defines the sampling edge of receiving data '0' = falling edge '1' = rising edge
8	0	**V_SD2_WAIT**	enables turn around cycles (wait cycles between FSC and first data bit) on SDO2 port '0' = disable '1' = enable
11..9	0	**V_BCLK2_WAIT**	defines the number of the coupling turn around cycles in BCCK1 clock units ('000' = 0 and so on)
12	0	**V_HW3_MASTER**	Defines the mode for PCM Highway 3. In slave mode the port FSC2 and BCLK2 are not driven by the HFC-S active (high-Z). '0' = slave '1' = master
13	0	**V_HW3_SD2_EN**	Enables permanently the SDO2 Port of the PCM Highway 2. If enabled the entry in the R_HW3_TS_EN has no influence. '0' = disable '1' = enable

Bits	Reset Value	Name	Description
14	0	**V_HW3_BUFF_OFF**	disables the switching buffer for the PCM Highway 3 '1' = off '0' = on
15		**(reserved)**	

R_HW_SL_CNT	**(read only)**	0x000B0210
Slot count register for the PCM highways		

Bits	Reset Value	Name	Description
4..0		**V_HW1_SL**	Number of the current time slot of highway 1
5		**V_HW1_BUFF**	buffer position of highway 1 switching buffer
7..6		**(reserved)**	
12..8		**V_HW2_SL**	Number of the current time slot of highway 2
13		**V_HW2_BUFF**	buffer position of highway 2 switching buffer
15..14		**(reserved)**	
20..16		**V_HW3_SL**	Number of the current time slot of highway 3
21		**V_HW3_BUFF**	buffer position of highway 3 switching buffer
31..22		**(reserved)**	

4.4 Switching unit

Table 34: *Overview of the* HFC-S active *switching unit registers*

Address	Name	Page	Address	Name	Page
0x000B00C0	R_HW1_IDX	120	0x000B01E4	R_ST_IDX	121
0x000B00E0	R_HW2_IDX	120	0x000B01E8	R_HW1_TS_EN	107
0x000B01C0	R_HW3_IDX	120	0x000B01EC	R_HW2_TS_EN	107
0x000B01E0	R_CODEC_IDX	121	0x000B01F0	R_HW3_TS_EN	108

The switching unit is a data distribution modul which can connect the PCM highways, the S/T interface and the CODEC module directly without keeping the ARM7^{TM} CPU busy. An overview of the data distribution is shown in figure 27. Alternative data streams can be processed via the CPU.

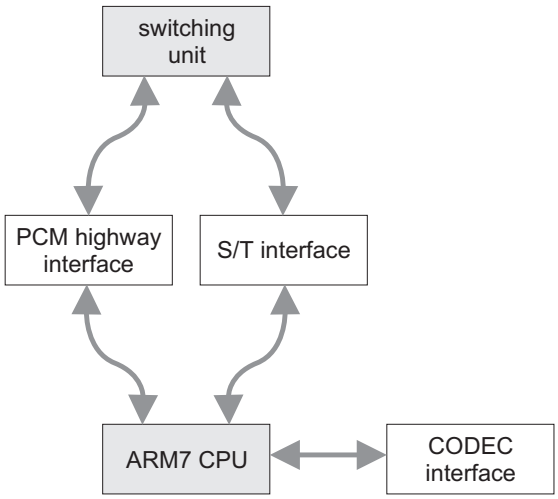


Figure 27: *Data distribution in the* HFC-S active *system*

4.4.1 Source index registers

The data coupling can be established between the 32 PCM time slots of each highway, the two CODECs (in compressed or linear data mode, one or two bytes each) and the B1- and B2-channels of the S/T interface. So there are $3 \times 32 + 2 \times 2 + 2 = 102$ data sources as listes in table 35. Each data source is implemented as an index register. Table 35 shows base addresses and index values for 32 bit (resp. 16 bit for the S/T interface) accesses. Alternatively, all registers can be accessed by word and byte access, e.g. address 0x000B00D1 to access time slot 17 of PCM highway 1.

4.4.2 Destination codes

For each data source the destination must be specified. The destination code is a 8 bit value. Table 36 shows all defined values. These have to be written into the data source index registers.

Table 35: *Source registers of the switching unit (TS = time slot, high (resp. low) = higher (resp. lower) byte of the CODECs 1 and 2)*

Data source	Source index register (base address + index)	Byte 3	Byte 2	Byte 1	Byte 0
32 time slots of PCM highway 1	R_HW1_IDX				
	0x000B00C0 + 0x00	TS 3	TS 2	TS 1	TS 0
	+ 0x04	TS 7	TS 6	TS 5	TS 4
	+ 0x08	TS 11	TS 10	TS 9	TS 8
	+ 0x0C	TS 15	TS 14	TS 13	TS 12
	+ 0x10	TS 19	TS 18	TS 17	TS 16
	+ 0x14	TS 23	TS 22	TS 21	TS 20
	+ 0x18	TS 25	TS 24	TS 23	TS 24
	+ 0x1C	TS 31	TS 30	TS 29	TS 28
32 time slots of PCM highway 2	R_HW2_IDX				
	0x000B00E0 + 0x00	TS 3	TS 2	TS 1	TS 0
	+ 0x04	TS 7	TS 6	TS 5	TS 4
	+ 0x08	TS 11	TS 10	TS 9	TS 8
	+ 0x0C	TS 15	TS 14	TS 13	TS 12
	+ 0x10	TS 19	TS 18	TS 17	TS 16
	+ 0x14	TS 23	TS 22	TS 21	TS 20
	+ 0x18	TS 25	TS 24	TS 23	TS 24
	+ 0x1C	TS 31	TS 30	TS 29	TS 28
32 time slots of PCM highway 3	R_HW3_IDX				
	0x000B01C0 + 0x00	TS 3	TS 2	TS 1	TS 0
	+ 0x04	TS 7	TS 6	TS 5	TS 4
	+ 0x08	TS 11	TS 10	TS 9	TS 8
	+ 0x0C	TS 15	TS 14	TS 13	TS 12
	+ 0x10	TS 19	TS 18	TS 17	TS 16
	+ 0x14	TS 23	TS 22	TS 21	TS 20
	+ 0x18	TS 25	TS 24	TS 23	TS 24
	+ 0x1C	TS 31	TS 30	TS 29	TS 28
CODEC	R_CODEC_IDX				
	0x000B01E0	high 2	low 2	high 1	low 1
S/T interface	R_ST_IDX				
	0x000B01E4	–	–	B2	B1

Table 36: *Destination codes of the switching unit (TS = time slot, high (resp. low) = higher (resp. lower) byte of the CODECs 1 and 2)*

HFC-S active **module**	**Data destination**	**Destination code**
PCM highway 1	32 time slots ($n = 0 \ldots 31$)	0x00 + n (range 0x00 ... 0x1F)
PCM highway 2	32 time slots ($n = 0 \ldots 31$)	0x20 + n (range 0x20 ... 0x3F)
PCM highway 3	32 time slots ($n = 0 \ldots 31$)	0x40 + n (range 0x40 ... 0x5F)
CODEC 1	low 1	0x60
	high 1	0x61
CODEC 2	low 2	0x62
	high 2	0x63
S/T interface	B1	0x64
	B2	0x65
–	'disable code'	0x80

4.4.3 Register description

R_HW1_IDX	(read / write)	0x000B00C0

Base address of the index register for the PCM highway 1.

The values written to this 32 byte register define the destination for the PCM data of each time slot.

Bits	Reset Value	Name	Description
31..0	0	V_HW1_IDX	the access to address 0x000B00C0 $+n \cdot 4$ selects the time slots $TS[n \cdot 4] \ldots TS[n \cdot 4 + 3]$ in byte $0 \ldots$ byte 3 ($n = 0 \ldots 7$)

Note: Values to be written into this register are shown in table 36

R_HW2_IDX	(read / write)	0x000B00E0

Base address of the index register for the PCM highway 2.

The values written to this 32 byte register define the destination for the PCM data of each time slot.

Bits	Reset Value	Name	Description
31..0	0	V_HW2_IDX	the access to address 0x000B00E0 $+n \cdot 4$ selects the time slots $TS[n \cdot 4] \ldots TS[n \cdot 4 + 3]$ in byte $0 \ldots$ byte 3 ($n = 0 \ldots 7$)

Note: Values to be written into this register are shown in table 36

R_HW3_IDX	(read / write)	0x000B01C0

Base address of the index register for the PCM highway 3.

The values written to this 32 byte register define the destination for the PCM data of each time slot.

Bits	Reset Value	Name	Description
31..0	0	V_HW3_IDX	the access to address 0x000B01C0 $+n \cdot 4$ selects the time slots $TS[n \cdot 4] \ldots TS[n \cdot 4 + 3]$ in byte $0 \ldots$ byte 3 ($n = 0 \ldots 7$)

Note: Values to be written into this register are shown in table 36

R_CODEC_IDX	(read / write)	0x000B01E0

Index register of the CODECs

Bits	Reset Value	Name	Description
7..0	0	**V_CODEC1L**	Lower byte for data coupling between CODEC and time slots of the PCM highways. The lower byte of the CODEC is used for the μ-law or a-law mode of the CODEC. In linear mode two time slots have to be used for 14 bit CODEC data (lower byte, higher byte).
15..8	0	**V_CODEC1H**	higher byte for data coupling between CODEC and time slots of the PCM highways
23..16	0	**V_CODEC2L**	lower byte for data coupling between CODEC and time slots of the PCM highways. The lower byte of the CODEC is used for the μ-law or a-law mode of the CODEC. In linear mode two time slots have to be used for 14 bit CODEC data (lower byte, higher byte).
31..24	0	**V_CODEC2H**	higher byte for data coupling between CODEC and time slots of the PCM highways

R_ST_IDX	(read / write)	0x000B01E4

Index register for the B1- and B2-channels of the S/T interface

Bits	Reset Value	Name	Description
7..0	0	**V_ST_B1**	coupling between B1-channel and time slots of the PCM highways
15..8	0	**V_ST_B2**	coupling between B2-channel and time slots of the PCM highways

4.5 CODEC module

4.5.1 Functional description

The HFC-S active has two sigma delta audio CODECs for speech and telephony applications.

Each CODEC contains both digital IIR / FIR filters and smoothing filters. The normal input and output channels have μ- / a-law format with 38 dB signal-to-noise distortion ratio. The digital data format for input and output data of this device can be 8 bit companded data (a-law, μ-law) or 14 bit linear data which can be easily selected by the CODEC control register.

An on-chip voltage reference circuit is included to allow single supply operation.

4.5.2 Register description

R_CODEC_TX	(read / write)	0x000D0000
Codec transmit register		

Bits	Reset Value	Name	Description
13..0	0	**V_CODEC1_TX**	data for CODEC 1 If companded data (a-law/μ-law) is selected, only bits 7 . . . 0 are used.
15..14	0	**(reserved)**	
29..16	0	**V_CODEC2_TX**	data for CODEC 2 If companded data (a-law/μ-law) is selected, only bits 7 . . . 0 are used.
31..30	0	**(reserved)**	

R_CODEC_RX	(read only)	0x000D0004
Codec linear mode receive register		

Bits	Reset Value	Name	Description
13..0	0	**V_CODEC1_RX**	data from CODEC 1 (valid in linear mode only)
15..14		**(reserved)**	reserved
29..16	0	**V_CODEC2_RX**	data from CODEC 2 (valid in linear mode only)
31..30		**(reserved)**	reserved

R_CODEC_RX8	(read only)	0x000D0008
Codec compand mode receive register		

Bits	Reset Value	Name	Description
7..0	0	**V_CODEC1_RX8**	data from CODEC 1 (valid in compand mode only)
15..8		**(reserved)**	
23..16	0	**V_CODEC2_RX8**	data from CODEC 2 (valid in compand mode only)
31..24		**(reserved)**	reserved

R_CODEC_CTRL	(read / write)	0x000D000C

Codec control register

Bits	Reset Value	Name	Description
0	0	**V_CODEC1_TX_LIN**	select CODEC 1 transmit mode '0' = 8 bit companded '1' = 14 bit linear
1	0	**V_CODEC2_TX_LIN**	select CODEC 2 transmit mode '0' = 8 bit companded '1' = 14 bit linear
2	0	**V_CODEC1_RX_LIN**	select CODEC 1 receive mode '0' = 8 bit companded '1' = 14 bit linear
3	0	**V_CODEC2_RX_LIN**	select CODEC 2 receive mode '0' = 8 bit companded '1' = 14 bit linear
4	0	**V_PCM_CODEC1**	use data from PCM highway as transmit data for CODEC 1 '0' = data from registers '1' = data from PCM highway
5	0	**V_PCM_CODEC2**	use data from PCM highway as transmit data for CODEC 2 '0' = data from registers '1' = data from PCM highway
6	0	**V_CODEC1_CODEC2**	use receive data from CODEC 1 as transmit data for CODEC 2 '0' = inactive '1' = active
7	0	**V_CODEC2_CODEC1**	use receive data from CODEC 2 as transmit data for CODEC 1. '0' = inactive '1' = active
8	0	**V_CODEC1_MUTE**	Codec 1 analog mute '1' = mute '0' = normal operation
9	0	**V_CODEC2_MUTE**	Codec 2 analog mute '1' = mute '0' = normal operation
10	0	**V_CODEC1_LOOP**	Codec 1 analog loopback '1' = loopback '0' = normal operation
11	0	**V_CODEC2_LOOP**	Codec 2 analog loopback '1' = loopback '0' = normal operation
12	0	**V_CODEC1_ADC_OFF**	Codec 1 ADC power down '1' = power down '0' = normal operation

Bits	Reset Value	Name	Description
13	0	**V_CODEC2_ADC_OFF**	Codec 2 ADC power down '1' = power down '0' = normal operation
14	0	**V_CODEC1_DAC_OFF**	Codec 1 DAC power down '1' = power down '0' = normal operation
15	0	**V_CODEC2_DAC_OFF**	Codec 2 DAC power down '1' = power down '0' = normal operation
16	0	**V_CODEC1_ALAW**	Codec 1 a-law/μ-law select, applies only when 8 bit companded data is selected '1'= a-law '0' = μ-law
17	0	**V_CODEC2_ALAW**	Codec 2 a-law/μ-law select, applies only when 8 bit companded data is selected '1' = a-law '0' = μ-law
25..18		**(reserved)**	must be set to 0x00
26	0	**V_CODEC1_RES**	controls reset signal of CODEC 1 '1' = reset '0' = normal operation
27	0	**V_CODEC2_RES**	controls reset signal of CODEC 2 '1' = reset '0' = normal operation
28	0	**V_CODEC1_INV**	invert data bits 6 … 0 for G.711 conformity in a-law mode, this bit should be set to '1' in a-law mode, else '0'
29	0	**V_CODEC2_INV**	invert data bits 6 … 0 for G.711 conformity in a-law mode, this bit should be set to '1' in a-law mode, else '0'
30	0	**V_CODEC1_INV7**	invert data bit 7 for G.711 conformity in μ-law mode, this bit should be set to '1' in μ-law mode, else '0'
31	0	**V_CODEC2_INV7**	invert data bit 7 for G.711 conformity in μ-law mode, this bit should be set to '1' in μ-law mode, else '0'.

5 Interfaces

5.1 General purpose input and output pins (GPIO)

5.1.1 Functional description

The HFC-S active has up to 31 GPIO pins:

- GPIO[15:0] : 16 bidirectional GPIO pins with interrupt capabilities
- GPIO[25:16] : 10 bidirectional GPIO pins
- GPI[0] : 1 GPI pin (input only)
- GPO[3:0] : 4 GPO pins (output only)

19 of these GPIOs are always mapped to package pins. The other 12 GPIOs can be mapped to other function pins when the primary function of those pins is not needed. It is possible to select the function (primary function / GPIO function) for each pin individually.

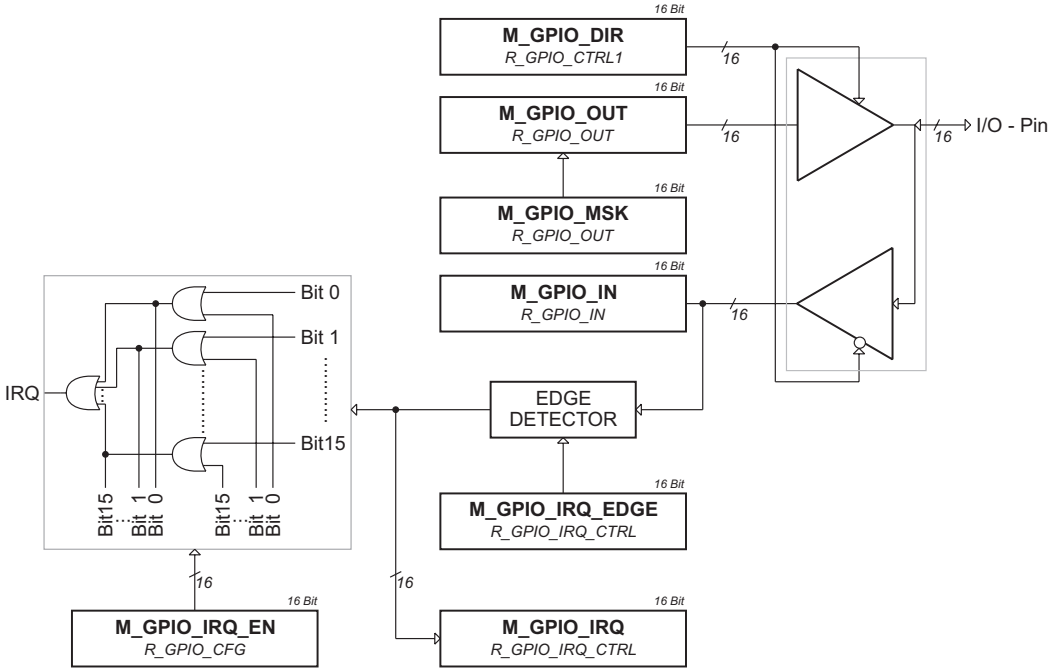


Figure 28: *Simplified representation for the primary GPIO[15:0] functionality*

The direction of most GPIOs is individually configurable by the software. Each port of GPIO[15:0] can be used as an input signal for the FSC-PLL synchronization. With these pins it is possible to output some internal signals (like time pulses, PWM pulses, FSC, watchdog pulse) on the GPIO ports. The interrupt capability of the GPIO[15:0] pins is also valid on these pins if the primary function is selected.

The bits of the data out register for GPIO[15:0] are maskable, i.e. that each bit can be set by the software without influence on any other bit. The edge for the GPIO interrupt is programmable for each bit of GPIO[15:0] and each GPIO interrupt can be enabled or disabled.

Table 37: *Overview of GPIO functions (DIR = signal direction input / output)*

Pin	Primary function		DIR	GPIO function	DIR
1	CARRY1	(timer 1 carry signal)	O	GPIO0	I / O
9	A20	(external address bus)	O	GPO2	O
10	A21	(external address bus)	O	GPO3	O
12	A18	(external address bus)	O	GPO0	O
13	A19	(external address bus)	O	GPO1	O
75				GPI0	I
86	CLK_OUT	(system clock f_{sys})	O	GPIO17	I / O
87				GPIO16	I / O
88	EOFT	(EOFT signal of the S/T interface)	O	GPIO15	I / O
89	DK_REP	(DK_REP signal of the S/T interface)	O	GPIO14	I / O
90	DK_EN	(DK_EN signal of the S/T interface)	O	GPIO13	I / O
101	SDI1	(serial data input for PCM highway 2)	I	GPIO18	I / O
102	SDO1	(serial data output for PCM highway 2)	O	GPIO19	I / O
103	BCLK1	(bit clock for PCM highway 2)	O	GPIO20	I / O
104	FSC1	(frame sync signal for PCM highway 2)	I / O	GPIO21	I / O
105	SDI2	(serial data input for PCM highway 3)	I	GPIO22	I / O
106	SDO2	(serial data output for PCM highway 3)	O	GPIO23	I / O
111	RXD	(serial receive data (UART))	I	GPIO24	I / O
112	TXD	(serial transmit data (UART))	O	GPIO25	I / O
115	CLK_ST	(S/T clock f_{ISDN})	O	GPIO12	I / O
116	CLK_EXT	(clock for external devices (f_{ext}))	O	GPIO11	I / O
117		()	O	GPIO10	I / O
118	FSC_TE	(FSC_TE signal of the S/T interface)	O	GPIO9	I / O
119	WDT	(carry signal of the watchdog timer)	O	GPIO8	I / O
120	PFS3	(peripheral frame sync 3 signal with interrupt capability)	O	GPIO7	I / O
121	PFS2	(peripheral frame sync 2 signal with interrupt capability)	O	GPIO6	I / O
122	PSF1	(peripheral frame sync 1 signal with interrupt capability)	O	GPIO5	I / O
157	PFS0	(peripheral frame sync 0 signal with interrupt capability)	O	GPIO4	I / O
158		()	O	GPIO3	I / O
159	PWM_OUT	(PWM output)	O	GPIO2	I / O
160	CARRY2	(Timer 2 carry signal)	O	GPIO1	I / O

5.1.2 Register description

R_GPIO_CFG	(read / write)	0x00090014

Configuration register for the primary GPIOs (interrupt and prescaler for FSC source)

Bits	Reset Value	Name	Description
15..0	0x0000	**V_GPIO_IRQ_EN**	interrupt enable register for each GPIO[15:0] pin each bit '1' = interrupt enable '0' = interrupt disable
25..16	0x000	**V_FSC_PREDIV**	predivider for the external FSC signal of GPIO[15:0] for the FSC-PLL synchronization The external FSC signal can be a multiple of 8 kHz
29..26	0	**V_FSC_GPIO_SEL**	Selects the GPIO port as source for the FSC synchronization One of 16 ports is selected
31..30		**(reserved)**	

R_GPIO_IRQ_CTRL	(read / write)	0x00090018

GPIO[15:0] interrupt control and status register

Bits	Reset Value	Name	Description
15..0	0x0000	**V_GPIO_IRQ**	shows the interrupt status for each GPIO interrupt (writing a zero value sets back the interrupt request)
31..16	0x0000	**V_GPIO_IRQ_EDGE**	selects the trigger edge for each GPIO interrupt '1' = rising edge '0' = falling edge

R_GPIO_OUT	(read / write)	0x0009001C

GPIO[15:0] data output register

Bits	Reset Value	Name	Description
15..0	0x0000	**V_GPIO_OUT**	sets the value (level) on each GPIO port. It takes one clock cycle to switch output data to this register.
31..16	0xFFFF	**V_GPIO_OUTMSK**	Sets the mask for the data output value Only a '1' in the mask allows new setting in the V_GPIO_OUT bitmap. '1' = on '0' = off

R_GPIO_IN1	(read only)	0x00090020

GPIO[15:0] data input register

Bits	Reset Value	Name	Description
15..0	0x0000	**V_GPIO_IN1**	represents the input status of the GPIO port.
31..16		**(reserved)**	

R_GPIO_IN2	(read only)	0x00080034

GPIO[25:16] and GPI[0] data input register

Bits	Reset Value	Name	Description
9..0		**V_GPIO_IN2**	represents the input status of the GPIO[25:16] port.
10		**V_GPI_IN**	represents the input status of the GPI[0] port.

R_GPIO_CTRL1	(read / write)	0x00090024

GPIO[15:0] output enable register and port mapping

Setting a bit of bitmap [31:16] disables the corresponding bit in V_GPIO_DIR ('0' = disable, '1' = enable)

Bits	Reset Value	Name	Description
15..0	0x0000	**V_GPIO_DIR**	controls the direction for each GPIO '1' = output '0' = input
16	0	**V_GPIO0_TI1**	enables timer 1 carry signal on GPIO 0
17	0	**V_GPIO1_TI2**	enables timer 2 carry signal on GPIO 1
18	0	**V_GPIO2_PWM**	enables PWM signal on GPIO 2
19	0	**V_GPIO3_FSC**	enables internal FSC signal on GPIO 3
20	0	**V_GPIO4_PFS0**	enables internal PFS0 (peripheral frame sync) signal on GPIO 4
21	0	**V_GPIO5_PFS1**	enables internal PFS1 (peripheral frame sync) signal on GPIO 5
22	0	**V_GPIO6_PFS2**	enables internal PFS2 (peripheral frame sync) signal on GPIO 6
23	0	**V_GPIO7_PFS3**	enables internal PFS3 (peripheral frame sync) signal on GPIO 7
24	0	**V_GPIO8_WD**	enables watchdog carry signal on GPIO 8
25	0	**V_GPIO9_FSC_ST**	enables the FSC_TE (S/T-Interface) signal on GPIO 9
26	0	**V_GPIO10_FSC_CONST**	enables the constructed FSC signal on GPIO 10
27	0	**V_GPIO11_CNT1B**	enables f_{ext} signal on GPIO 11
28	0	**V_GPIO12_CNT1A**	enables internal S/T interface clock (12.288/2 MHz) signal on GPIO 12
29	0	**V_GPIO13_DKEN**	enables the internal DK_EN signal of the S/T interface on GPIO 13
30	0	**V_GPIO14_DKREP**	enables the internal DK_REP signal of the S/T interface on GPIO 14
31	0	**V_GPIO15_EOFT**	enables the internal EOFT signal of the S/T interface on GPIO 15

R_GPO_CTRL	(read / write)	0x0008002C

GPO[3:0] control register

Bits	Reset Value	Name	Description
3..0	0	**V_GPO_DATA**	4 bit output value for GPO[3:0] (applies only when the corresponding bit of V_GPO0_EN ... V_GPO3_EN is set)
4	0	**V_GPO0_EN**	selects function of pin ADDR[18] '1' = secondary function GPO[0] '0' = ADDR[18]
5	0	**V_GPO1_EN**	selects function of pin ADDR[19] '1' = secondary function GPO[1] '0' = ADDR[19]
6	0	**V_GPO2_EN**	selects function of pin ADDR[20] '1' = secondary function GPO[2] '0' = ADDR[20]
7	0	**V_GPO3_EN**	selects function of pin ADDR[21] '1' = secondary function GPO[3] '0' = ADDR[21]
31..8		**(reserved)**	

R_GPIO_CTRL2	()	0x00080030

GPIO[25:16] output control register and port mapping

Bits	Reset Value	Name	Description
9..0	0x000	**V_GPIO_DATA**	10 bit output value on GPIO[25:16]
19..10	0x000	**V_GPIO_EN**	enable for GPIO[25:16] '1' = data output '0' = 'HIGH-Z' **Note:** Enable must be set for primary and secondary function to obtain output functionality.
20	0	**V_GPIO18_EN**	selects function of pin SDI1 '1' = secondary function GPIO[18] '0' = SDI1
21	0	**V_GPIO19_EN**	selects function of pin SDO1 '1' = secondary function GPIO[19] '0' = SDO1
22	0	**V_GPIO20_EN**	selects function of pin BCLK1 '1' = secondary function GPIO[20] '0' = BCLK1
23	0	**V_GPIO21_EN**	selects function of pin FSC1 '1' = secondary function GPIO[21] '0' = FSC1
24	0	**V_GPIO22_EN**	selects function of pin SDI2 '1' = secondary function GPIO[22] '0' = SDI2
25	0	**V_GPIO23_EN**	selects function of pin SDO2 '1' = secondary function GPIO[23] '0' = SDO2
26	0	**V_GPIO24_EN**	selects function of pin RxD '1' = secondary function GPIO[24] '0' = RxD
27	0	**V_GPIO25_EN**	selects function of pin TxD '1' = secondary function GPIO[25] '0' = TxD
28	0	**V_NCTS_EN**	selects function of pin BCLK2 '1' = secondary function NCTS '0' = BCLK2
29	0	**V_NRTS_EN**	selects function of pin FSC2 '1' = secondary function NRTS '0' = FSC2
30	0	**V_GPIO17_CNT1B**	enables the f_{ext} clock output on GPIO[17] '1' = clock on GPIO17 '0' = GPIO[17]
31		**(reserved)**	

5.2 UART module

5.2.1 Functional description

The HFC-S active contains an *Universal Asynchronous Receiver / Transmitter* (UART) module with 8 byte FIFO for both directions. The complete logic is on chip to minimize system overhead and to maximize system efficiency. The UART module performs a serial-to-parallel conversion on data characters received from a peripheral device (e.g. PC) or a modem, and parallel-to-serial conversion on data characters received from the ARM7^{TM} CPU. The ARM7^{TM} CPU can read the complete status of the UART module at any time during the functional operation. The reported status information includes the type and condition of the transfer operations being performed by the UART module as well as any error conditions (parity, overrun, framing or break interrupt). The UART module includes a programmable baud rate generator that is capable of dividing the timing reference clock input f_{sys} to achieve a baud rate

$$f_{baud} = \frac{f_{sys}}{8 \cdot \text{V\_UART\_BAUD}}$$

within the scope V_UART_BAUD = 1 ... 216.

The UART has complete handshake control capability and a processor interrupt system. Interrupts can be programmed to the users requirements which minimizes the CPU time required to handle the communications link. The UART module is always running on system clock speed f_{sys}.

Features:

- Transmitter and receiver are each buffered with 8 byte FIFOs to reduce the number of interrupts.
- Adds or removes standard asynchronous communication bits (start, stop and parity) to or from the serial data.
- Independently controlled interrupts for transmit, receive, line status and FIFO status
- Programmable baud rate generator divides the system clock f_{sys}
- Handshake control functions (CTS, RTS) with enable / disable functionality
- Fully programmable serial interface characteristics
  - 5-, 6-, 7- or 8-bit characters
  - Even, odd or no-parity bit generation and detection
  - 1-, 1/2- or 2-stop bit generation
  - Baud rate generation (up to 6 Mbaud)
- Complete status reporting capabilities
- Internal diagnostic capabilities:
  - Loop back controls for communications link fault isolation
  - Break, parity, overrun and framing error simulation
- Fully prioritized interrupt system controls

The physical UART interface operates with 3.3 V input and output voltage levels. For the connection to a RS 232 interface an external line driver (e.g. MAX 560) is required. The polarity of the RXD and TXD ports are programmable to active low or active high logic levels. Figure 29 illustrates the signal form (voltage level) depending on the output mode.

Table 38: *Baud rate programming values (* : rounded value)*

Baud rate	f_{sys}	V_UART_BAUD	Baud rate	f_{sys}	V_UART_BAUD
	12.288 MHz	13964 *		12.288 MHz	5120
	24.576 MHz	27927 *		24.576 MHz	10240
110	36.864 MHz	41891 *	300	36.864 MHz	15360
	49.152 MHz	55855 *		49.152 MHz	20480
	61.440 MHz	69818 *		61.440 MHz	25600
	12.288 MHz	1280		12.288 MHz	640
	24.576 MHz	2560		24.576 MHz	1280
1200	36.864 MHz	3840	2400	36.864 MHz	1920
	49.152 MHz	5120		49.152 MHz	2560
	61.440 MHz	6400		61.440 MHz	3200
	12.288 MHz	320		12.288 MHz	160
	24.576 MHz	640		24.576 MHz	320
4800	36.864 MHz	960	9600	36.864 MHz	480
	49.152 MHz	1280		49.152 MHz	640
	61.440 MHz	1600		61.440 MHz	800
	12.288 MHz	80		12.288 MHz	40
	24.576 MHz	160		24.576 MHz	80
19200	36.864 MHz	240	38400	36.864 MHz	120
	49.152 MHz	320		49.152 MHz	160
	61.440 MHz	400		61.440 MHz	200
	12.288 MHz	27 *		12.288 MHz	13 *
	24.576 MHz	53 *		24.576 MHz	27 *
57600	36.864 MHz	80	115200	36.864 MHz	40
	49.152 MHz	107 *		49.152 MHz	53 *
	61.440 MHz	133 *		61.440 MHz	67 *
	12.288 MHz	7 *		12.288 MHz	3 *
	24.576 MHz	13 *		24.576 MHz	7 *
230400	36.864 MHz	20	460800	36.864 MHz	10
	49.152 MHz	27 *		49.152 MHz	13 *
	61.440 MHz	33 *		61.440 MHz	17 *
	12.288 MHz	2 *			
	24.576 MHz	3 *			
921600	36.864 MHz	5			
	49.152 MHz	7 *			
	61.440 MHz	8 *			

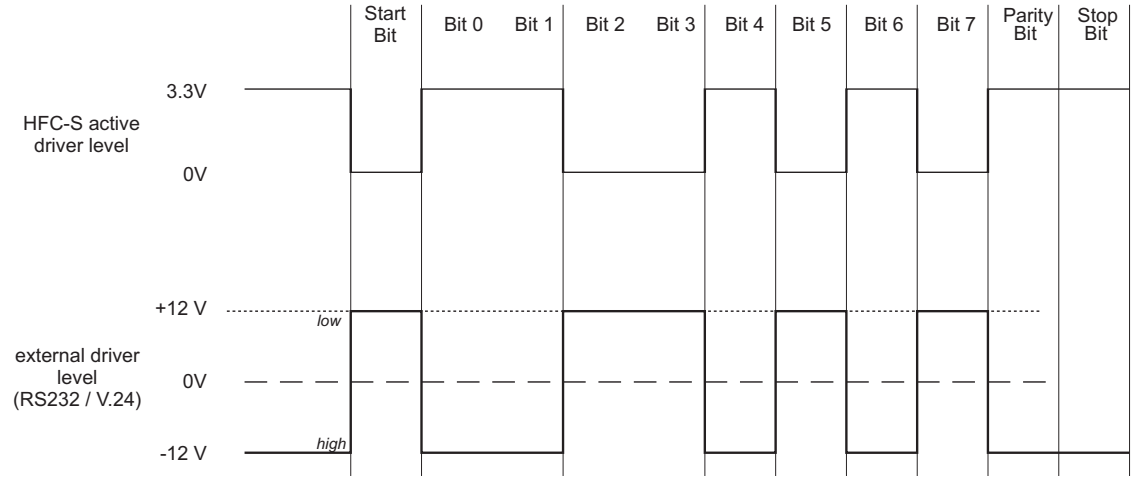


Figure 29: *Logic levels of the UART interface*

5.2.2 Register description

R_UART_TX1	(write only)	0x000A0000

Transmit register of the UART interface (Register value cannot be read back)

Note: The R_UART_RX1 register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_TX1**	Basic transmit register. The software should write this register only for transmitting.
15..9		**(reserved)**	

R_UART_TX2	(write only)	0x000A0004

Transmit register of the UART interface (Register value cannot be read back)

Note: The R_UART_RX2 register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_TX2**	2nd transmit register This register can be used for multiple transmission (multiple write command)
15..9		**(reserved)**	

R_UART_TX3	(write only)	0x000A0008

Transmit register of the UART interface (Register value cannot be read back)

Note: The R_UART_RX3 register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_TX3**	3rd transmit register This register can be used for multiple transmission (multiple write command)
15..9		**(reserved)**	

R_UART_TX4	(write only)	0x000A000C

Transmit register of the UART interface (Register value cannot be read back)

Note: The R_UART_RX4 register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_TX4**	4th transmit register This register can be used for multiple transmission (multiple write command)
15..9		**(reserved)**	

R_UART_BAUD	(write only)	0x000A0010

Register for baud rate settings

Bits	Reset Value	Name	Description
15..0	0x001B	**V_UART_BAUD**	Programmable baud rate UART baud rate $= \frac{f_{sys}}{8 \cdot \text{V\_UART\_BAUD}}$

R_UART_CFG	(write only)	0x000A0020

Configuration register for the UART module
Note: The R_UART_PREVIEW register is located on the same address

Bits	Reset Value	Name	Description
1..0	0	**V_UART_STB**	defines the length of the stop bit '0x' = 1 stop bit '10' = 1.5 stop bits '11' = 2 stop bits
4..2	0	**V_UART_PAR**	defines the parity mode of the parity bit '000' = no parity bit '001' = even parity '010' = odd parity '011' = mark '100' = space
9..5	0x08	**V_UART_LEN**	defines the word length of the data (e.g. '0111' = 7 bit)
10	0	**V_UART_CTS**	CTS signal '0'= CTS not used '1'= CTS used
14..11	6	**V_RX_HWA**	Sets the high water mark for interrupt generation of received data. An interrupt is generated when the number of received bytes is equal to this value.
18..15	1	**V_UART_TX_LWA**	Sets the low water mark for interrupt generation of transmit data. An interrupt is generated when the number of the transmit FIFO data is equal to this value.
19	0	**V_UART_TX_POL**	defines polarity of the TxD output '0' = non inverted '1' = inverted
20	0	**V_UART_RX_POL**	defines polarity of the RxD input '0' = non inverted '1' = inverted
31..21		**(reserved)**	

R_UART_CLR	(write only)	0x000A0024

Clears the UART FIFOs

Note: The R_UART_STATUS register is located on the same address

Bits	Reset Value	Name	Description
0	0	**V_UART_CLR**	clears the UART FIFO
7..1		**(reserved)**	

R_UART_ECHO	(read / write)	0x000A0028

Sets the UART into an echo mode
The received data is transmitted by the hardware immediately

Bits	Reset Value	Name	Description
0	0	**V_UART_ECHO**	enables the UART echo mode '0' = hardware echo off '1' = hardware echo on
7..1		**(reserved)**	

R_UART_RX1	(read only)	0x000A0000

Receive register of the UART interface

Note: The R_UART_TX1 register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_RX1**	Basic receive register The software should read this register only for receiving
15..9		**(reserved)**	
19..16	0	**V_UART_TX1_FIFO**	number of bytes in the transmit FIFO
23..20	0	**V_UART_RX1_FIFO**	number of bytes in the receive FIFO
24	0	**V_UART_PERR1**	parity error information of V_UART_RX1 '0' = no parity error '1' = parity error
31..25		**(reserved)**	

R_UART_RX2	(read only)	0x000A0004

Receive register of the UART interface

Note: R_UART_TX2 register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_RX2**	2nd receive register This register can be used for multiple receiving (multiple read command)
15..9		**(reserved)**	
19..16	0	**V_UART_TX2_FIFO**	number of bytes in the transmit FIFO
23..20	0	**V_UART_RX2_FIFO**	number of bytes in the receive FIFO
24	0	**V_UART_PERR2**	parity error information of V_UART_RX2 '0' = no parity error '1' = parity error
31..25		**(reserved)**	

R_UART_RX3	(read only)	0x000A0008

Receive register of the UART interface

Note: R_UART_TX3 register is located on the same address

Bits	Reset Value	Name	Description
7..0	0x00	**V_UART_RX3**	3rd receive register This register can be used for multiple receiving (multiple read command)
15..8		**(reserved)**	
19..16	0	**V_UART_TX3_FIFO**	number of bytes in the transmit FIFO
23..20	0	**V_UART_RX3_FIFO**	number of bytes in the receive FIFO
24	0	**V_UART_PERR3**	parity error information of V_UART_RX3 '0' = no parity error '1' = parity error
31..25		**(reserved)**	

R_UART_RX4	(read only)	0x000A000C

Receive register of the UART interface

Note: R_UART_TX4 register is located on the same address

Bits	Reset Value	Name	Description
7..0	0x00	**V_UART_RX4**	Fourth receive register This register can be used for multiple receiving (multiple read command)
15..8		**(reserved)**	
19..16	0	**V_UART_TX4_FIFO**	number of bytes in the transmit FIFO
23..20	0	**V_UART_RX4_FIFO**	number of bytes in the receive FIFO
24	0	**V_UART_PERR4**	parity error information of V_UART_RX4 '0' = no parity error '1' = parity error
31..25		**(reserved)**	

R_UART_PREVIEW	(read only)	0x000A0020

Previous status register for the UART module, for preview the next FIFO data

Note: The R_UART_CFG register is located on the same address

Bits	Reset Value	Name	Description
8..0	0x000	**V_UART_PREVIEW**	shows the next data in the receive FIFO
15..9		**(reserved)**	
19..16	0	**V_UART_TX0_FIFO**	number of bytes in the transmit FIFO
23..20	0	**V_UART_RX0_FIFO**	number of bytes in the receive FIFO
24	0	**V_UART_PERR0**	parity error information of V_UART_PREVIEW '0' = no parity error '1' = parity error
31..25		**(reserved)**	

Table 39: *Bitmap description of the UART transmit FIFO status*

Bit number	Bit name	Description
0	status	number of words in the transmit FIFO
1	low_water	low water mark of the transmit FIFO has been reached
2	empty	transmit FIFO is empty (last byte sent)

Table 40: *Bitmap description of the UART receive FIFO status*

Bit number	Bit name	Description
0	status	words received in the FIFO
1	high_water	hiwh water mark of the receive FIFO has been reached
2	full	receive FIFO is full
3	parity_error	parity error in the receive FIFO
4	echo_error	echo error
5	break	Hardware echo error (TxD and RxD at the same time)
6	overflow	overflow of the receive FIFO (data lost)

R_UART_STATUS **(read only)** 0x000A0024

Interrupt Status register for the UART module

Note: The R_UART_CLR register is located on the same address

Bits	Reset Value	Name	Description
3..0	0	**V_UART_TX_FIFO**	Number of bytes in the transmit FIFO
7..4	0	**V_UART_RX_FIFO**	Number of bytes in the receive FIFO
10..8	0	**V_UART_TX_STATUS**	status of send data (data sent out)
17..11	0	**V_UART_RX_STATUS**	status of receive data (data received)
32..18		**(reserved)**	

Note: see table 39 (page 143) and 40 (page 143) for bit identification of V_UART_TX_STATUS and V_UART_RX_STATUS

R_UART_IRQ_CFG	(read / write)	0x000A002C

Interrupt status and configuration register for the UART interface.

Bits	Reset Value	Name	Description
2..0	0	**V_UART_TX_IRQ_STATUS**	interrupt status of send data (data sent out)
9..3	0x00	**V_UART_RX_IRQ_STATUS**	interrupt status of receive data (data received)
12..10	0	**V_UART_TX_FIQ_EN**	enables fast interrupts for data transmit
19..13	0x00	**V_UART_RX_FIQ_EN**	enables fast interrupts for data receive
22..20	0	**V_UART_TX_IRQ_EN**	enables interrupts for transmit data
29..23	0	**V_UART_RX_IRQ_EN**	enables interrupts for receive data
31..30		**(reserved)**	

Note: see table 39 (page 143) and 40 (page 143) for bit identification of V_UART_TX_IRQ_STATUS and V_UART_RX_IRQ_STATUS

5.3 USB module

> **Note !**
>
> Please contact our support team if you want to use the USB interface.

Table 41: *Overview of the* HFC-S active *USB pins*

Number	Name	Description
73	USB+	differential USB port (positive)
74	USB-	differential USB port (negative)

Table 42: *Overview of the* HFC-S active *USB registers*

Address	Name	Page	Address	Name	Page
0x0008003C	R_USB_DRV	147	0x000E0030	R_USB_IEP_SEL	153
0x000E0000	R_USB_ADDR	147	0x000E0034	R_USB_IDATA	153
0x000E0004	R_USB_CFG	148	0x000E0038	R_USB_ICMD	154
0x000E0008	R_USB_CTRL	148	0x000E003C	R_USB_ISTATUS	154
0x000E000C	R_USB_EV1	149	0x000E0040	R_USB_OEP_EN	155
0x000E0010	R_USB_EVMSK1	150	0x000E0044	R_USB_IEP_EN	155
0x000E0014	R_USB_EV2	150	0x000E0048	R_USB_OEP_STALL	156
0x000E0018	R_USB_EVMSK2	151	0x000E004C	R_USB_IEP_STALL	156
0x000E0020	R_USB_OEP_SEL	151	0x000E0050	R_USB_OEP_EV	157
0x000E0024	R_USB_ODATA	151	0x000E0054	R_USB_OEP_EVMSK	157
0x000E0028	R_USB_OCMD	152	0x000E0058	R_USB_IEP_EV	157
0x000E002C	R_USB_OSTATUS	152	0x000E005C	R_USB_IEP_EVMSK	158

The HFC-S active has a complete Universal Serial Bus (USB) interface which is compatible with the USB specification 1.1. The on-chip USB transceiver permits the direct connection to the physical USB interface (e.g. computers with USB interface).

Four endpoints excluding endpoint 0 are implemented for transmit and receive direction. The endpoints 1 ... 4 have an FIFO depth of 64 byte for each direction, whereas endpoint 0 has an 16 byte FIFO for transmit and receive each.

The HFC-S active USB module supports control, interrupt and bulk transfer types.

Figure 30 illustrated the input and output driver circuitry of the USB interface. The receivers can be switched off with V_USB_OFF = 1 (register R_OSC_CFG). The differential and single ended data can be monitored in the named register bits of the register R_USB_DRV.

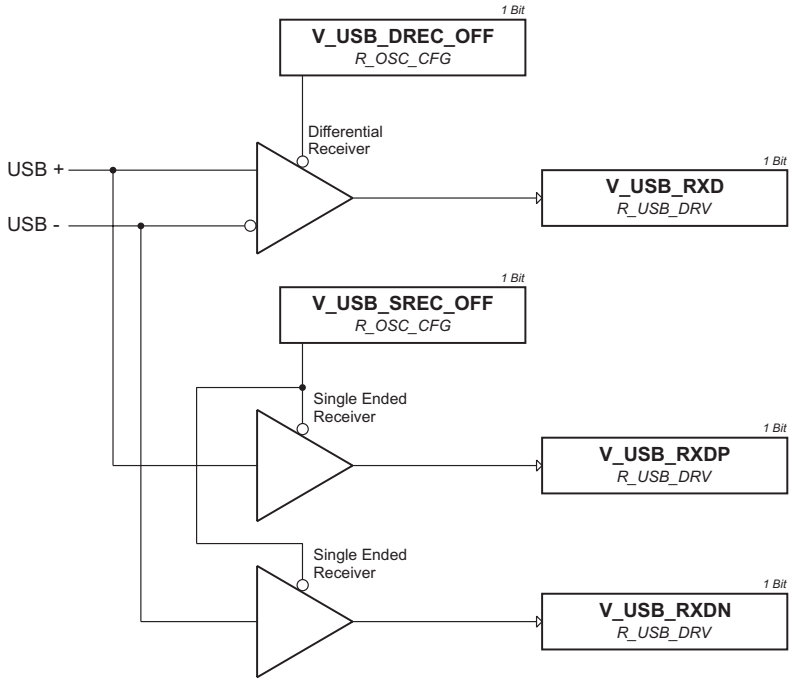


Figure 30: *USB input scheme*

5.3.1 Register description

R_USB_DRV	(read / write, read)	0x0008003C
USB driver control register		

Bits	Reset Value	Name	Description
0	1	**V_USB_WAK**	Generates a wakeup signal
1		**V_USB_RXD**	Represents the state of the RXD port
2		**V_USB_RXDP**	Represents the state of the RXDP port
3		**V_USB_RXDN**	Represents the state of the RXDN port
7..4		**(reserved)**	

R_USB_ADDR	(read / write)	0x000E0000
USB device address		

Bits	Reset Value	Name	Description
6..0	0x00	**V_USB_ADDR**	After power-on-reset the device works on the default address 0. The software has to set the new address after the status stage following a SET ADDRESS request directed to the USB device.
7		**(reserved)**	

R_USB_CFG	(read / write, read)	0x000E0004

USB mode configuration register

Bits	Reset Value	Name	Description
0	0	**V_USB_IRQ_POL**	Defines the interrupt polarity (0 = active low, 1 = active high)
1	0	**V_USB_IRQ_EN**	Interrupt output enable '1' = Enable the interrupt output. If one or more event bits of the R_USB_EV1 register are set before enabling the interrupt, an interrupt will also occure after setting this bit.
7..2		**V_USB_VER**	Represents the version of the USB module

R_USB_CTRL	(read / write)	0x000E0008

USB control register

Bits	Reset Value	Name	Description
0	0	**V_USB_RESU**	Force resume Setting this bit forces a K state on the data lines driven for 12 ms. If this bit is set immediately after a suspend condition, the resume will not start earlier than 5 ms after this suspend state.
1	0	**V_USB_AT**	Node attached When set to 0, the data lines will held on low to simulate a disconnected device. After initialisation of the microcontroller this bit should be set to 1 to force the recognition of a connect.
2	0	**V_USB_SUSP**	Suspend mode This bit must be set before entering the suspend state. Any bus activity starts the resume sequence and the bit will be cleared after its completion automatically. If no resume is performed, the bit must be cleared manually by the microcontroller.
7..3		**(reserved)**	

R_USB_EV1	(read only)	0x000E000C
1st USB event register		

Bits	Reset Value	Name	Description
0	0	**V_USB_EV2**	Event on the 2nd USB event register '1' = A change in the R_USB_EV2 register occurred. This leads to an interrupt if the corresponding bit in the R_USB_EVMSK2 register is set.
1		**(reserved)**	
2	0	**V_USB_ZOF**	ZERO OUT function '1' = An OUT token with an empty DATA1 packet was received on EP0. In this special case the FIFO is not blocked.
3		**(reserved)**	
4	0	**V_USB_IEV**	IN event function '1' = A bit in the R_USB_IEP_EV register changed to '1' and the corresponding bit in the R_USB_IEP_EVMSK register is set. This shows, that an event on an IN endpoint occurred and leads to an interrupt if the corresponding bit in the R_USB_EVMSK1 register is set.
5	0	**V_USB_OEV**	OUT event function '1' = A bit in the R_USB_OEP_EV register changed to '1' and the corresponding bit in the R_USB_OEP_EVMSK register is set. This shows, that an event on an OUT endpoint occurred and leads to an interrupt if the corresponding bit in the R_USB_EVMSK1 register is set.
7..6		**(reserved)**	

R_USB_EVMSK1	(read / write)	0x000E0010

Mask for the 1st USB event register

Bits	Reset Value	Name	Description
0	0	**V_USB_EVMSK2**	EVENT2 mask
1		**(reserved)**	
2	0	**V_USB_ZOFMSK**	ZERO OUT function mask
3		**(reserved)**	
4	0	**V_USB_IEVMSK**	IN event function mask
5	0	**V_USB_OEVMSK**	OUT event function mask
7..6		**(reserved)**	

R_USB_EV2	(read only)	0x000E0014

2nd USB event register

Bits	Reset Value	Name	Description
0	0	**V_USB_RES**	USB reset '1' = An USB reset is recognized at the data lines.
1	0	**V_USB_SUSP**	Suspend '1' = No data traffic was seen for 3 ms. The firmware should enter the suspend state if this event occures.
2	0	**V_USB_RESU**	Resume '1' = A resume state at the data lines was detected.
3	0	**V_USB_RWAK**	Remote wakeup '1' = A remote wakeup signal was detected. The bit can only be set, if the corresponding mask bit is set and the suspend mode is enabled using the V_USB_SUSP bit of the R_USB_CTRL register. If remote wakeup capability is enabled, the device must send a resume (setting bit V_USB_RESU of the R_USB_CTRL register) to wakeup the entire bus.
4	0	**V_USB_RDY**	Resume completed '1' = A resume is completed by the host with a low-speed EOP
7..5		**(reserved)**	

R_USB_EVMSK2	(read / write)	0x000E0018

Mask for the 2nd USB event register

Bits	Reset Value	Name	Description
0	0	**V_USB_RESMSK**	USB reset mask
1	0	**V_USB_SUSPMSK**	Suspend mask
2	0	**V_USB_RESUMSK**	Resume mask
3	0	**V_USB_RWAKMSK**	Remote wakeup mask
4	0	**V_USB_RDYMSK**	Resume completed mask
7..5		**(reserved)**	

R_USB_OEP_SEL	(read / write)	0x000E0020

OUT endpoint select register

Bits	Reset Value	Name	Description
3..0	0	**V_USB_OEP_SEL**	Writing the appropriate endpoint number allows the access to the mapped FIFO control and data registers (range 0 … 4).
7..4		**(reserved)**	

R_USB_ODATA	(read only)	0x000E0024

Data OUT register

Bits	Reset Value	Name	Description
7..0		**V_USB_ODATA**	The FIFO data of the selected endpoint can be read out by reading this register n times (n = value of V_USB_OCNT in the V_USB_OCNT register)
10..8		**(reserved)**	

R_USB_OCMD	(read / write)	0x000E0028

OUT command register

Bits	Reset Value	Name	Description
0		**V_USB_FLUSH**	Flush OUT FIFO The FIFO must be flushed by the microcontroller by setting this bit to '1'.
7..1		**(reserved)**	

Important !

If the bit V_USB_FLUSH is not set, the FIFO is blocked and the USB module will respond with a NAK handshake to OUT tokens. Because SETUP tokens dont have to be NAKed, the USB module will not respond to these tokens if the FIFO is blocked and the USB host will recognize a timeout.

R_USB_OSTATUS	(read only)	0x000E002C

OUT status register

Bits	Reset Value	Name	Description
6..0	0	**V_USB_OCNT**	OUT count Shows the number of received bytes residing in the output FIFO of the selected endpoint (EP0: range 0 … 16, EP1…4: range 0 … 64).
7	0	**V_USB_SETUP**	SETUP token received 1 = A SETUP transfer occurred on the endpoint 0.

R_USB_IEP_SEL	(read / write)	0x000E0030

IN endpoint select register

Bits	Reset Value	Name	Description
2..0	0	**V_USB_IEP_SEL**	Writing the appropriate endpoint number allows the access to the mapped FIFO control and data registers (range 0 … 4)).
7..3		**(reserved)**	

R_USB_IDATA	(write only)	0x000E0034

Data IN register

Bits	Reset Value	Name	Description
7..0		**V_USB_IDATA**	Data written to this register will enter the input FIFO selected by the R_USB_IEP_SEL register. The pointers for the FIFO will be automatically incremented with every write cycle and will be resetted if the input FIFO was flushed. The number of bytes contained in a FIFO is visible in the R_USB_ISTATUS register.

R_USB_ICMD	(read / write)	0x000E0038

IN command register

Bits	Reset Value	Name	Description
0	0	**V_USB_IFLUSH**	Flush IN FIFO The FIFO will be flushed automatically after a successful transmission of data to the host and receiving the appropriate ACK handshake. The FIFO can be flushed manually by setting this bit. The full- or empty-status of the FIFO can be evaluated by reading this bit.
1	0	**V_USB_IN_EN**	In transfer enable Setting this bit enables the USB module to sent the contents of the FIFO after the next IN token to this endpoint. The bit is cleared by hardware after a successful transmission. Any necessary retries caused by missing or destroyed handshake tokens will be done automatically.
7..2		**(reserved)**	

R_USB_ISTATUS	(read only)	0x000E003C

IN status register

Bits	Reset Value	Name	Description
6..0	0	**V_USB_ICNT**	IN count Shows the actual number of bytes in the selected input FIFO (EP0: range 0 … 16, EP1…4: range 0 … 64).
7		**V_USB_TOG**	IN toggle value Shows the actual toggle value of the DATA PID for the transfer (0 = DATA0 packet, 1 = DATA1 packet). The toggle value will be generated automatically.

R_USB_OEP_EN	(read / write, read)	0x000E0040

OUT endpoint enable register

If a bit is set, the device will respond to an OUT token addressed to the corresponding endpoint. If the corresponding endpoint FIFO is empty, the following data packet will be received and an ACK token will be sent in the handshake phase. Otherwise a NAK handshake will be sent to force a retry by the host. This register should be modified according to the endpoint layout chosen with the SET CONFIGURATION request.

Bits	Reset Value	Name	Description
0		**V_USB_OEP0_EN**	Endpoint 0 This bit is set automatically after detection of an USB Reset
4..1		**V_USB_OEP_EN**	Endpoints 1 … 4
7..5		**(reserved)**	

R_USB_IEP_EN	(read / write, read)	0x000E0044

IN endpoint enable register

If a bit is set, the function will respond to an IN token addressed to the corresponding endpoint. If the corresponding endpoint FIFO contains data and is enabled, the FIFO contents will be sent, otherwise a NAK handshake will be sent. This register should be modified according to the endpoint layout chosen with the SET CONFIGURATION request.

Bits	Reset Value	Name	Description
0		**V_USB_IEP0_EN**	Endpoint 0 This bit is set automatically after detection of an USB Reset
4..1		**V_USB_IEP_EN**	Endpoints 1 … 4
7..5		**(reserved)**	

R_USB_OEP_STALL	(read / write)	0x000E0048

OUT endpoint stall register

If a bit is set, the corresponding endpoint will always sent a STALL token during the handshake phase of an OUT transfer. SETUP transfers on control endpoint EP0 can not be stalled. The bit EP0 works only for OUT tokens on EP0 but not for SETUP tokens! (For example the OUT data stage after a SET DESCRIPTOR request) The bit for control endpoint EP0 will be automatically resetted after detection of a valid SETUP token. A bit of this register can only be set after the corresponding bit in the R_USB_OEP_EN register was set, otherwise the bit will stuck at zero.

Bits	Reset Value	Name	Description
4..0	0x00	**V_USB_OEP_STALL**	Enable OUT endpoint stall (range: 0 … 4)
7..5		**(reserved)**	

R_USB_IEP_STALL	(read / write)	0x000E004C

IN endpoint stall register

If a bit is set, the corresponding endpoint will always sent a STALL token after the data phase of an IN transfer. The bit for control endpoint EP0 will be automatically resetted after detection of a valid SETUP token. A bit of this register can only be set after the corresponding bit in the R_USB_IEP_EN register was set, otherwise the bit will stuck at zero.

Bits	Reset Value	Name	Description
4..0	0x00	**V_USB_IEP_STALL**	Endpoints 0 … 4
7..5		**(reserved)**	

R_USB_OEP_EV	(read only)	0x000E0050

OUT endpoint event register

The bits are set, when an OUT transfer on the corresponding endpoint is completed. Any set of a bit will force a set of the V_USB_OEV bit in the R_USB_EV1 register when the corresponding bit in the R_USB_OEP_EVMSK register is set. All bits are cleared on read.

Bits	Reset Value	Name	Description
4..0	0x00	**V_USB_OEP_EV**	Endpoits 0 … 4
7..5		**(reserved)**	

R_USB_OEP_EVMSK	(read / write)	0x000E0054

Mask of the OUT endpoint event register

Bits	Reset Value	Name	Description
4..0	0x00	**V_USB_OEP_EVMSK**	Endpoints 0 … 4
7..5		**(reserved)**	

R_USB_IEP_EV	(read / write)	0x000E0058

IN endpoint event register

The bits are set, when an IN transfer on the corresponding endpoint is completed. Any set of a bit will force a set of the V_USB_IEV bit in the R_USB_EV1 register when the corresponding bit in the R_USB_IEP_EVMSK register is set. All bits are cleared on read.

Bits	Reset Value	Name	Description
4..0	0x00	**V_USB_IEP_EV**	Endpoints 0 … 4
7..5		**(reserved)**	

R_USB_IEP_EVMSK	(read / write)	0x000E005C

Mask of the IN endpoint event register

Bits	Reset Value	Name	Description
4..0		**V_USB_IEP_EVMSK**	Endpoints 0 … 5
7..5		**(reserved)**	

A HFC-S active **package dimensions**

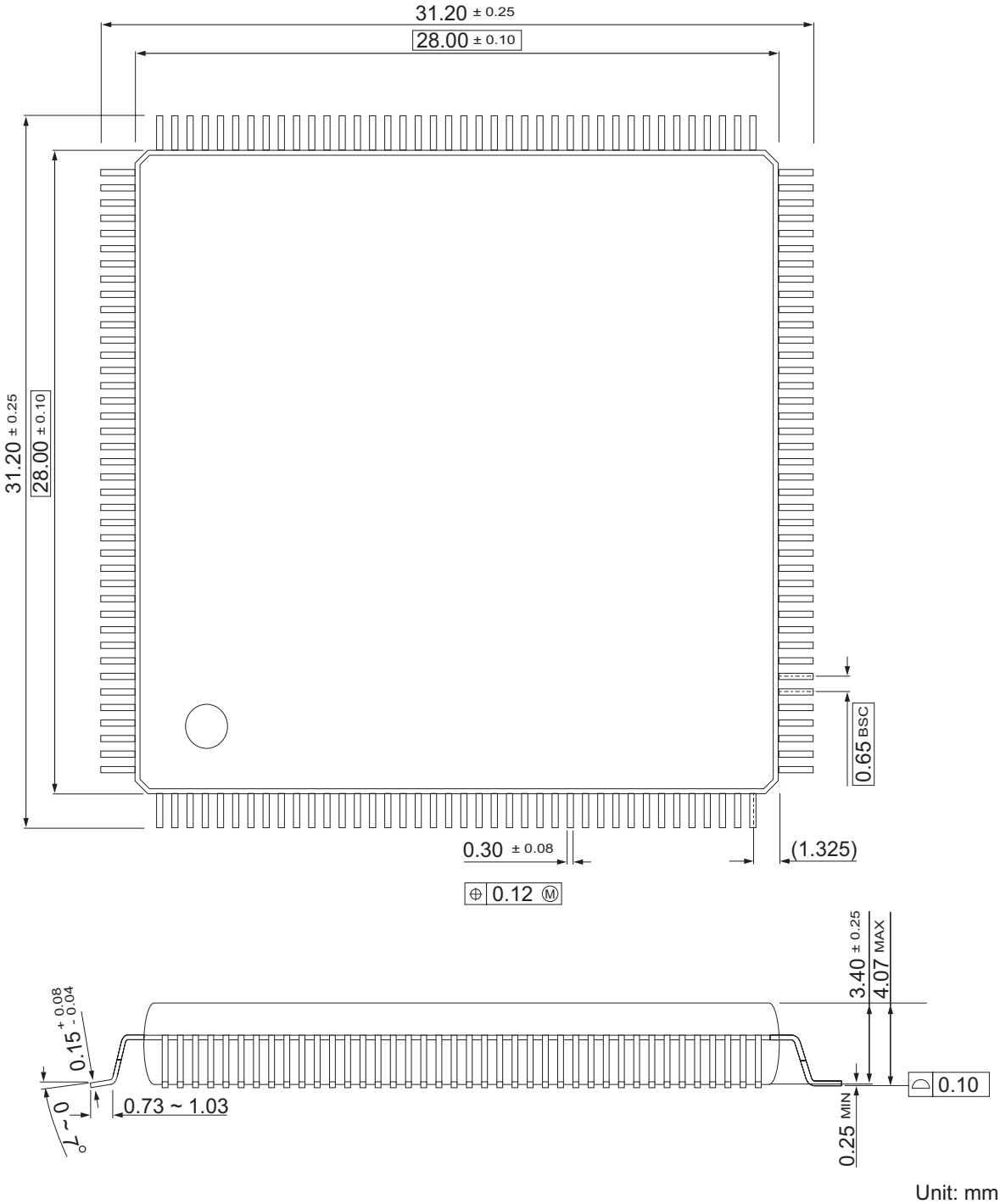


Figure 31: HFC-S active *package dimensions*

B Power supply and ground distribution

B.1 Digital supply pins

All ditigal power supply pins and digital ground pins listed in table 43 should be connected to the 3.3 V system supply voltage respectively to ground.

Table 43: *Power supply and ground pins of the digital subsystems*

Category	Pin name	Pin numbers
digital ground	DGND	18, 24, 30, 41, 55, 61, 72, 91, 108, 114
digital power supply	DVCC	8, 21, 29, 42, 58, 71, 92, 107, 113

B.2 Analog supply pins

Table 44: *Power supply and ground pins of the mixed signal subsystems*

Category	PLL 1 pin number (name)	PLL 2 pin number (name)	CODEC 1 pin number (name)	CODEC 2 pin number (name)
digital power supply	149 (DVCC_PLL1)	130 (DVCC_PLL2)	139 (DVCC_CODEC)	
digital ground	154 (DGND)	125 (DGND)	140 (DGND)	
analog power supply	150 (AVCC_PLL1)	129 (AVCC_PLL2)	148 (AVCC_CODEC1)	131 (AVCC_CODEC2)
analog ground	152 (AGND) 153 (AGND)	126 (AGND) 127 (AGND)	145 (AGND)	134 (AGND)

C Multiplexer control logic of the PLL 1 block

The PLL and the divider can be selected or bridged by multiplexers individually. One multiplexer can select the PLL output (V_PLL1_SEL = 1) or the original oscillator frequency (V_PLL1_SEL = 0). As the clock phase of the PLL output relative to the PLL input is uncontrollable, it is not possible to use a normal asynchronous multiplexer for this purpose. A normal multiplexer could generate very short clock pulses (spikes) during switching, resulting in misbehavior of the ARM7^{TM} CPU or flipflops in the circuit. Instead, a special logic has been developed to control a multiplexer in a way that it will switch from one input to zero level before switching to the other input, switching only when the individual input is at zero level (see fig. 32).

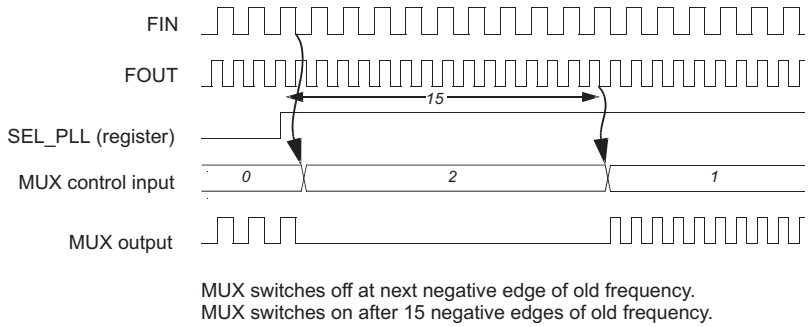


Figure 32: *Clock switching behaviour of PLL multiplexer*

The result is a multiplexer output waveform which contains no spikes or dynamic hazards, even though a pause in the output waveform is unavoidable for a versatile multiplexer that can also be used for switching between clock frequencies differing by a large factor.

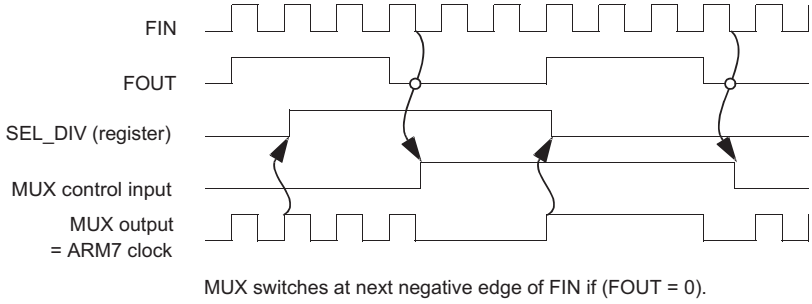
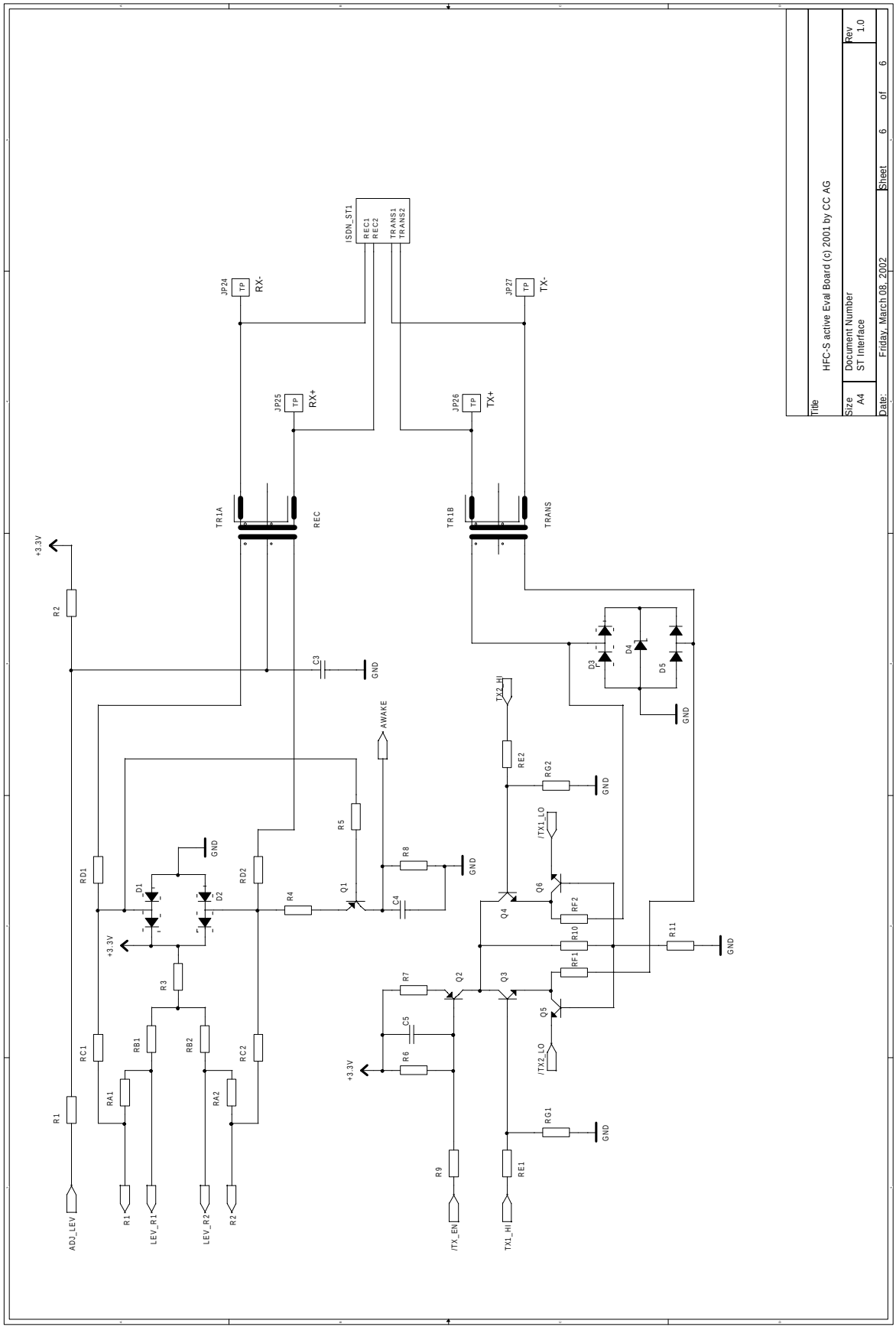


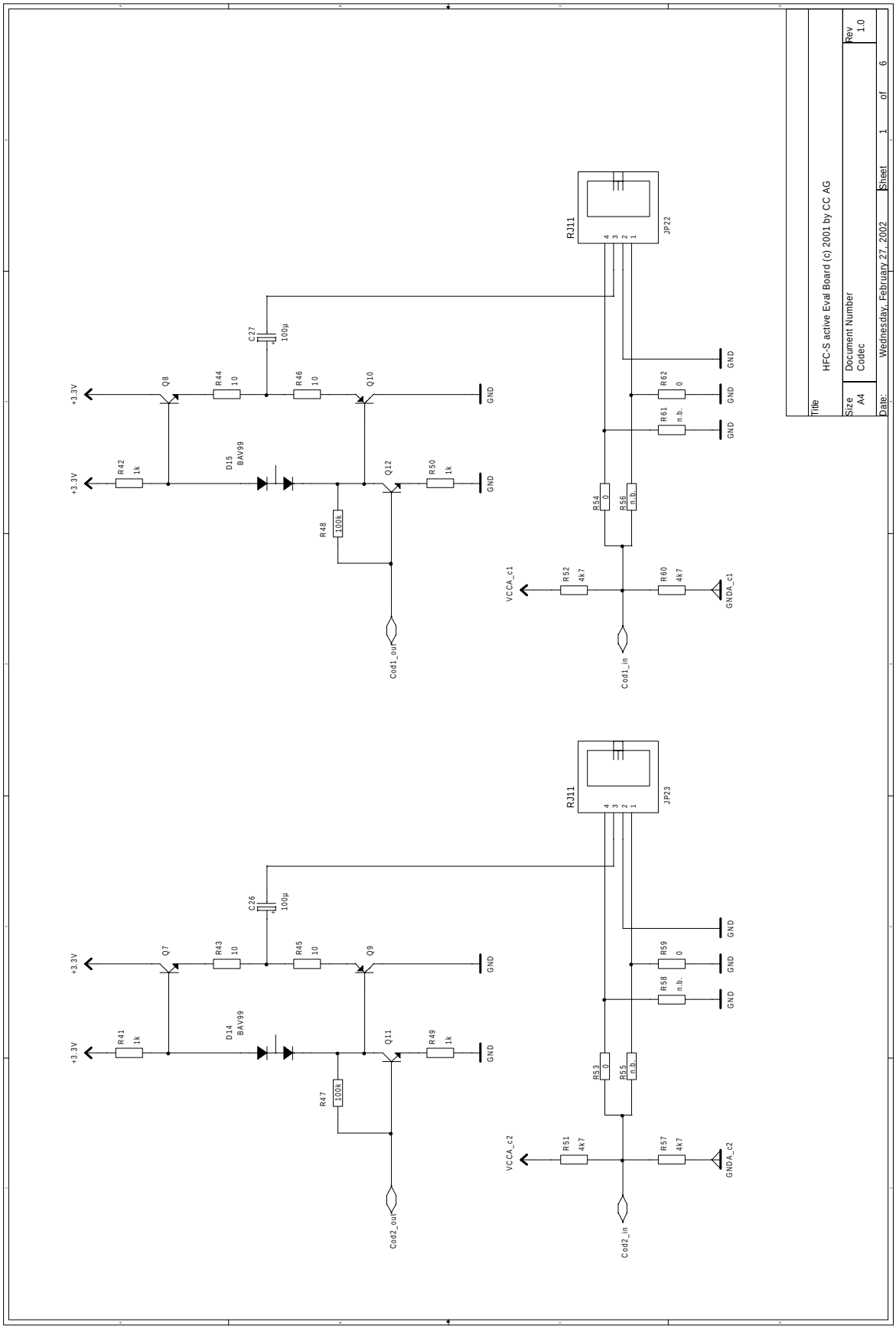
Figure 33: *Clock switching behaviour of divider multiplexer*

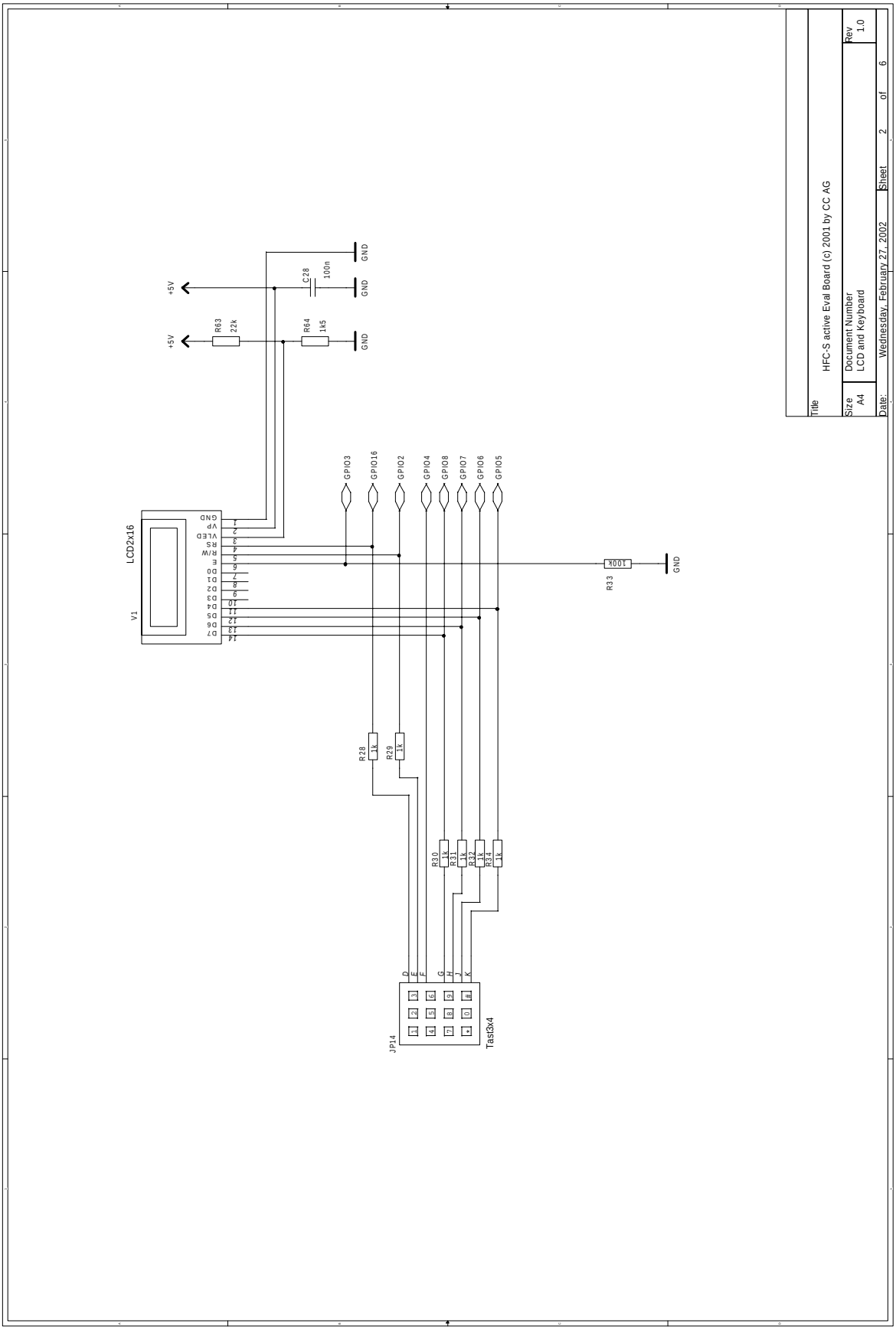
As the phase relationship between the divider input and the divider output is well defined in this case, the multiplexer can be controlled so that it switches directly from one input to the other at a time when both inputs are at zero level as shown in the diagram above.

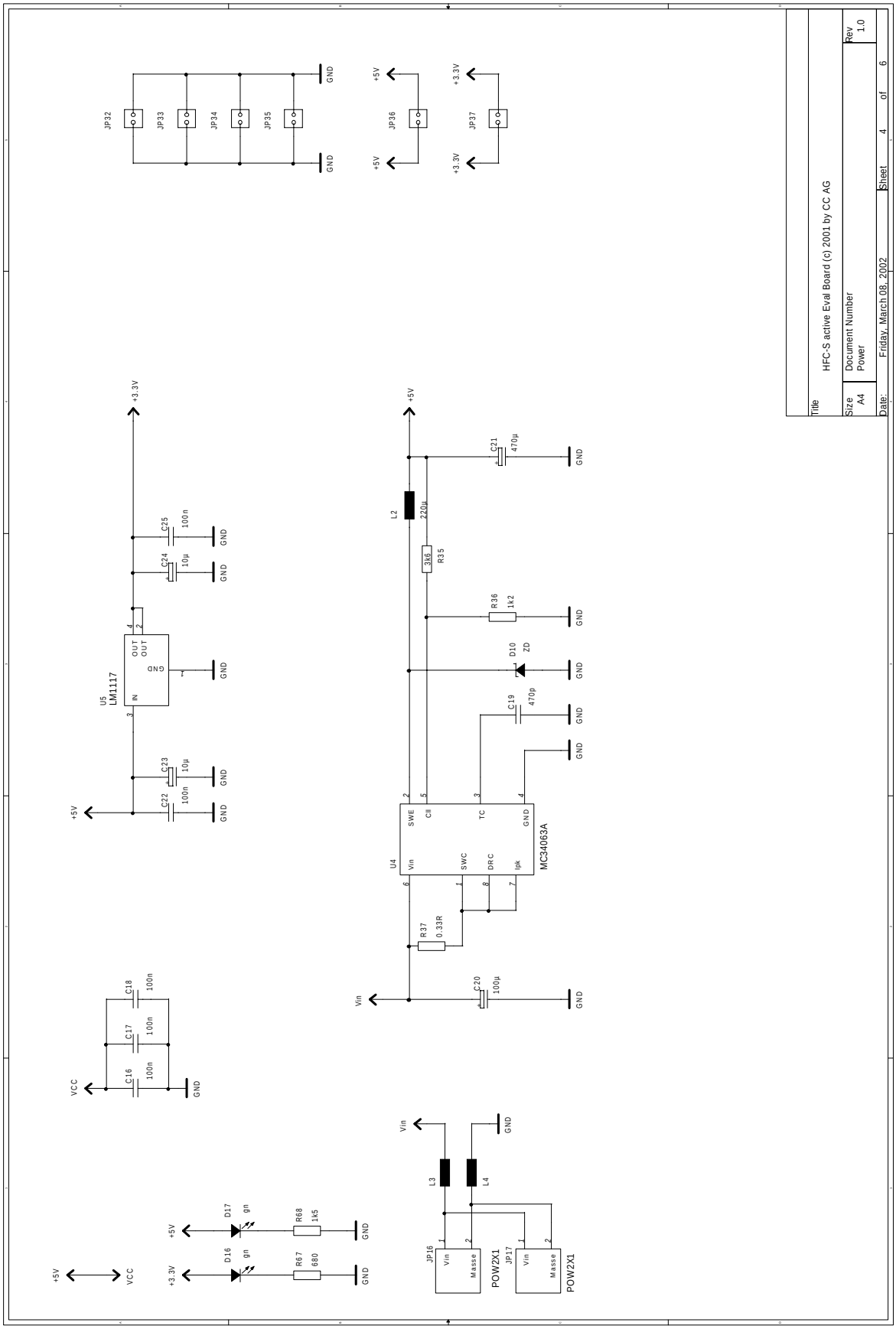
D Examples circuitry for HFC-S active

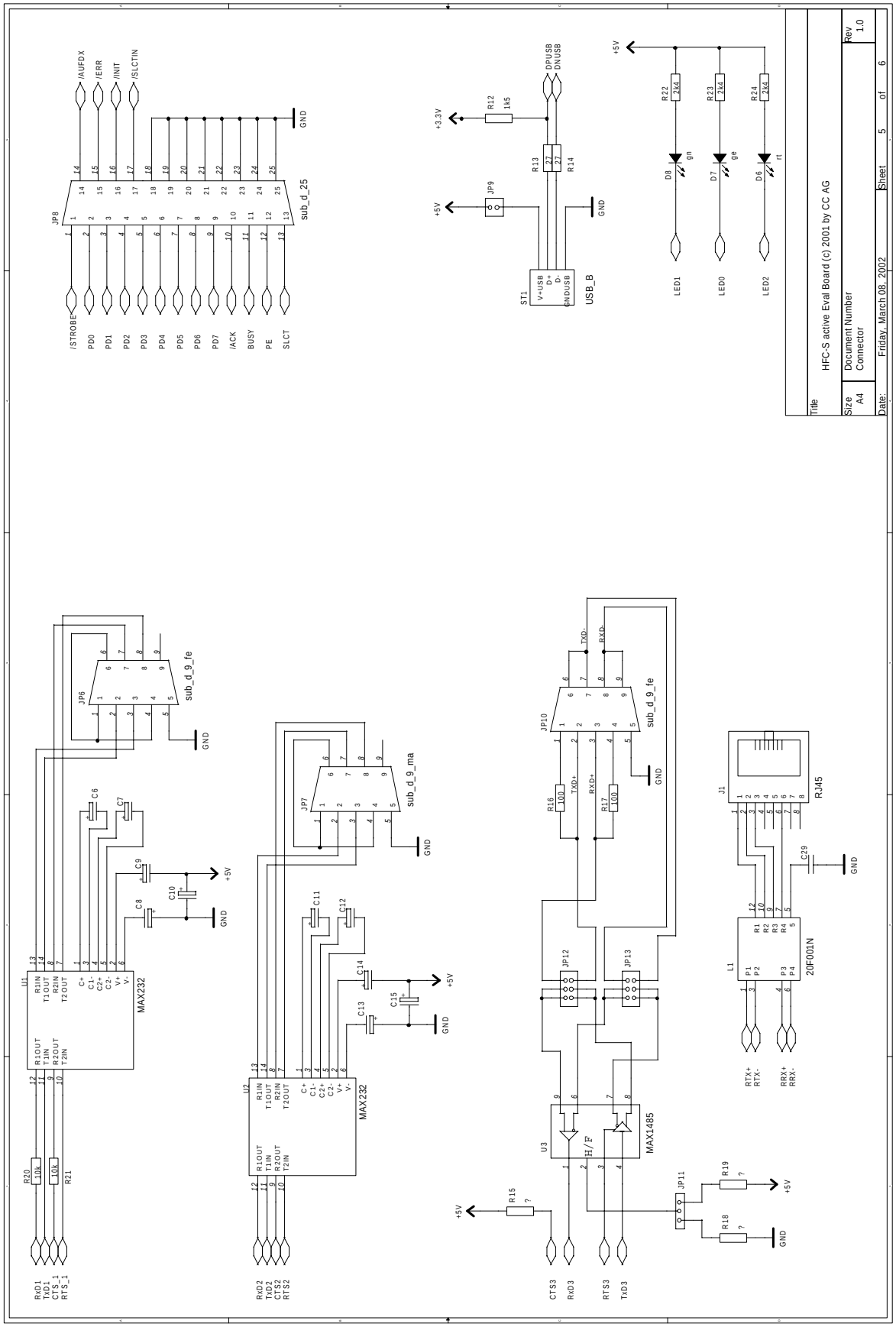
The following pages show the Cologne Chip evaluation board circuitry of the HFC-S active.

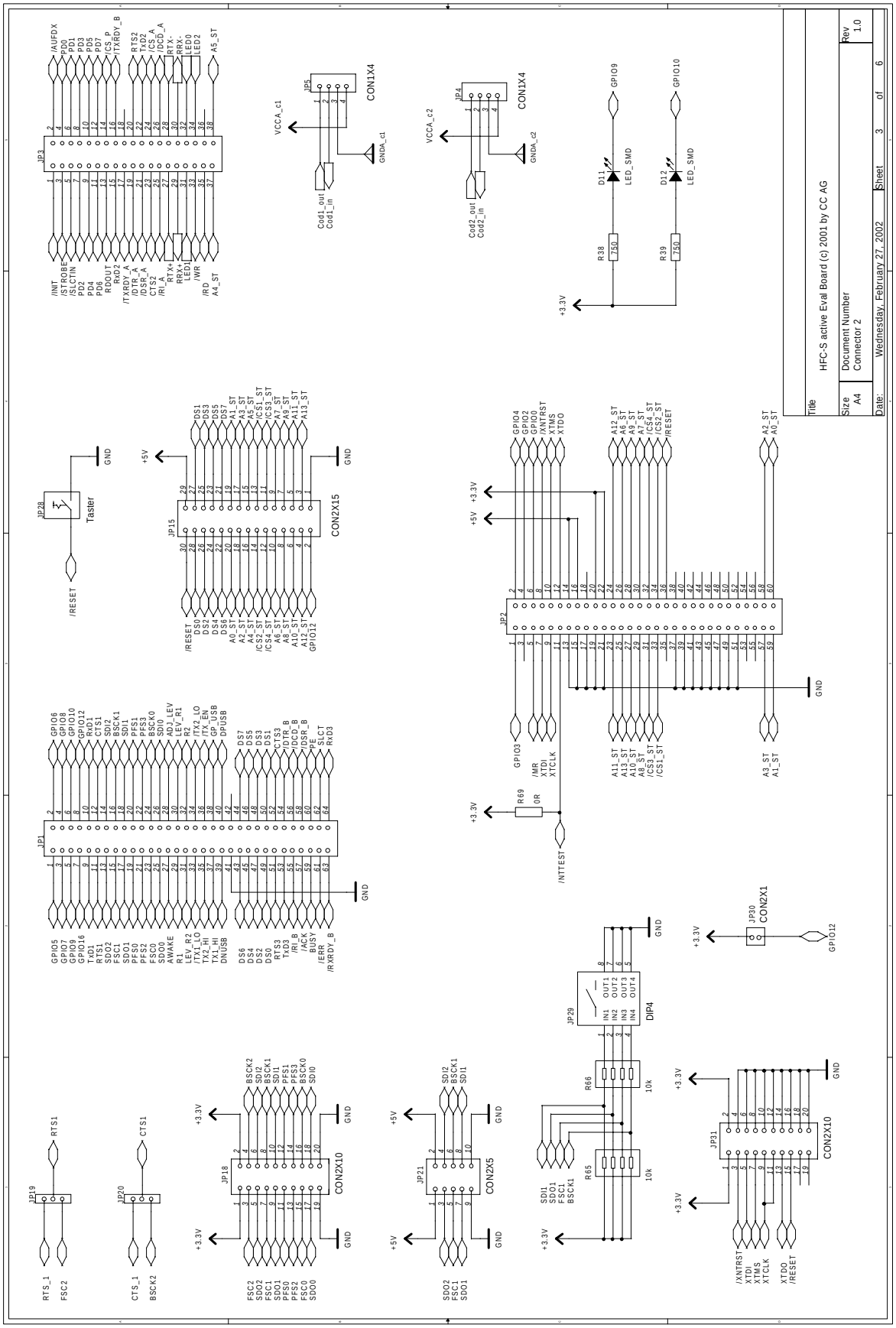

ISDN_ST1
REC1
REC2
TRANS1
TRANS2
JP24
RX-
JP25
RX+
JP26
TX+
JP27
TX-
TR1A
REC
TR1B
TRANS
+3.3V
R2
C3
GND
AWAKE
TX2_HI
RE2
RG2
/TX1_LO
R5
R8
RD1
RD2
D1
D2
R4
Q1
C4
Q6
Q4
RF2
R10
R11
RF1
R3
R7
Q2
Q3
Q5
C5
R6
/TX2_LO
RC1
RB1
RB2
RC2
RA1
RA2
R1
RG1
R9
RE1
ADJ_LEV
LEV_R1
LEV_R2
/TX_EN
TX1_HI
Title
HFC-S active Eval Board (c) 2001 by CC AG
Size
A4
Document Number
ST Interface
Rev
1.0
Date:
Friday, March 08, 2002
Sheet
6
of
6

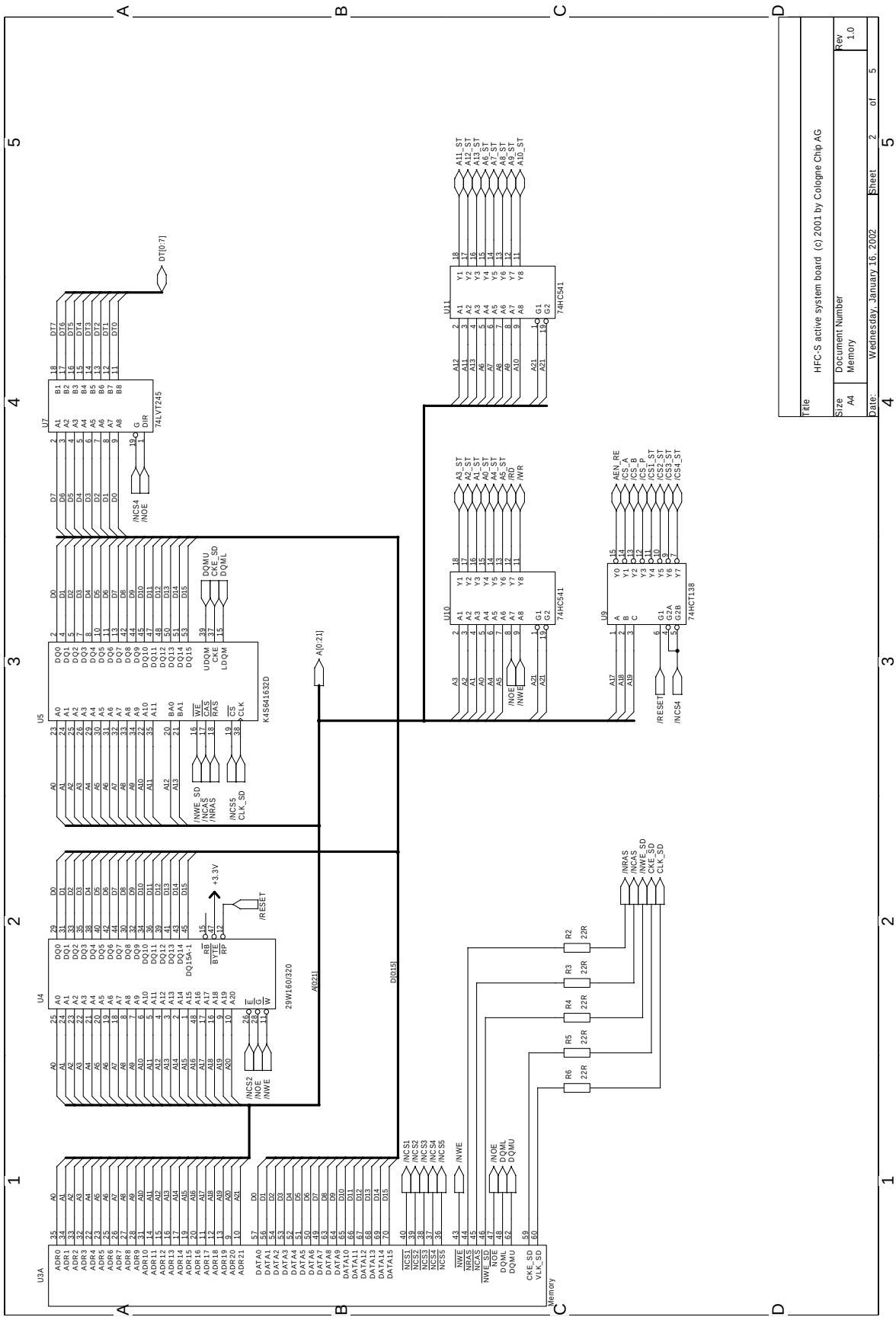

Title
HFC-S active Eval Board (c) 2001 by CC AG
Size
A4
Document Number
Codec
Rev
1.0
Date:
Wednesday, February 27, 2002
Sheet
1
of
6
RJ11
JP22
JP23
C27
C26
100µ
Q8
Q10
Q7
Q9
Q12
Q11
R44
R46
R43
R45
R42
R50
R41
R49
R48
R47
100k
D15
D14
BAV99
+3.3V
GND
R62
R61
R59
R58
R54
R56
R53
R55
n.b.
R52
R60
R51
R57
4k7
VCCA_c1
GNDA_c1
VCCA_c2
GNDA_c2
Cod1_out
Cod1_in
Cod2_out
Cod2_in

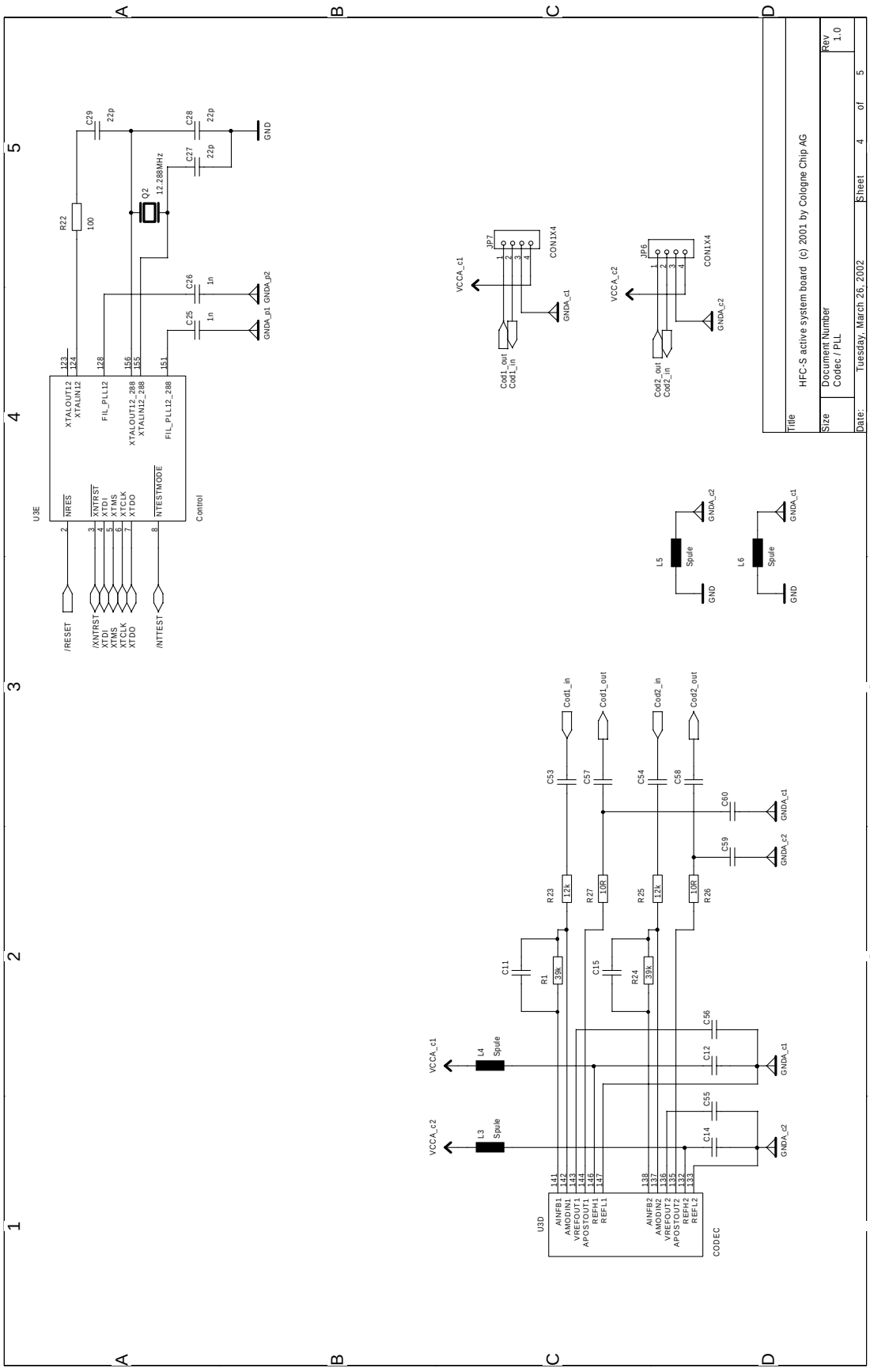

HFC-S active Eval Board (c) 2001 by CC AG
Document Number
LCD and Keyboard
Rev 1.0
Sheet 2 of 6
Wednesday, February 27, 2002
Size A4
Date:
Title
LCD2x16
V1
GND
VP
VLED
RS
R/W
E
D0
D1
D2
D3
D4
D5
D6
D7
GPIO3
GPIO16
GPIO2
GPIO4
GPIO8
GPIO7
GPIO6
GPIO5
+5V
C28 100n
R63 22k
R64 1k5
R33 100K
R28 1k
R29 1k
R30 1k
R31 1k
R32 1k
R34 1k
JP14
Tast3x4

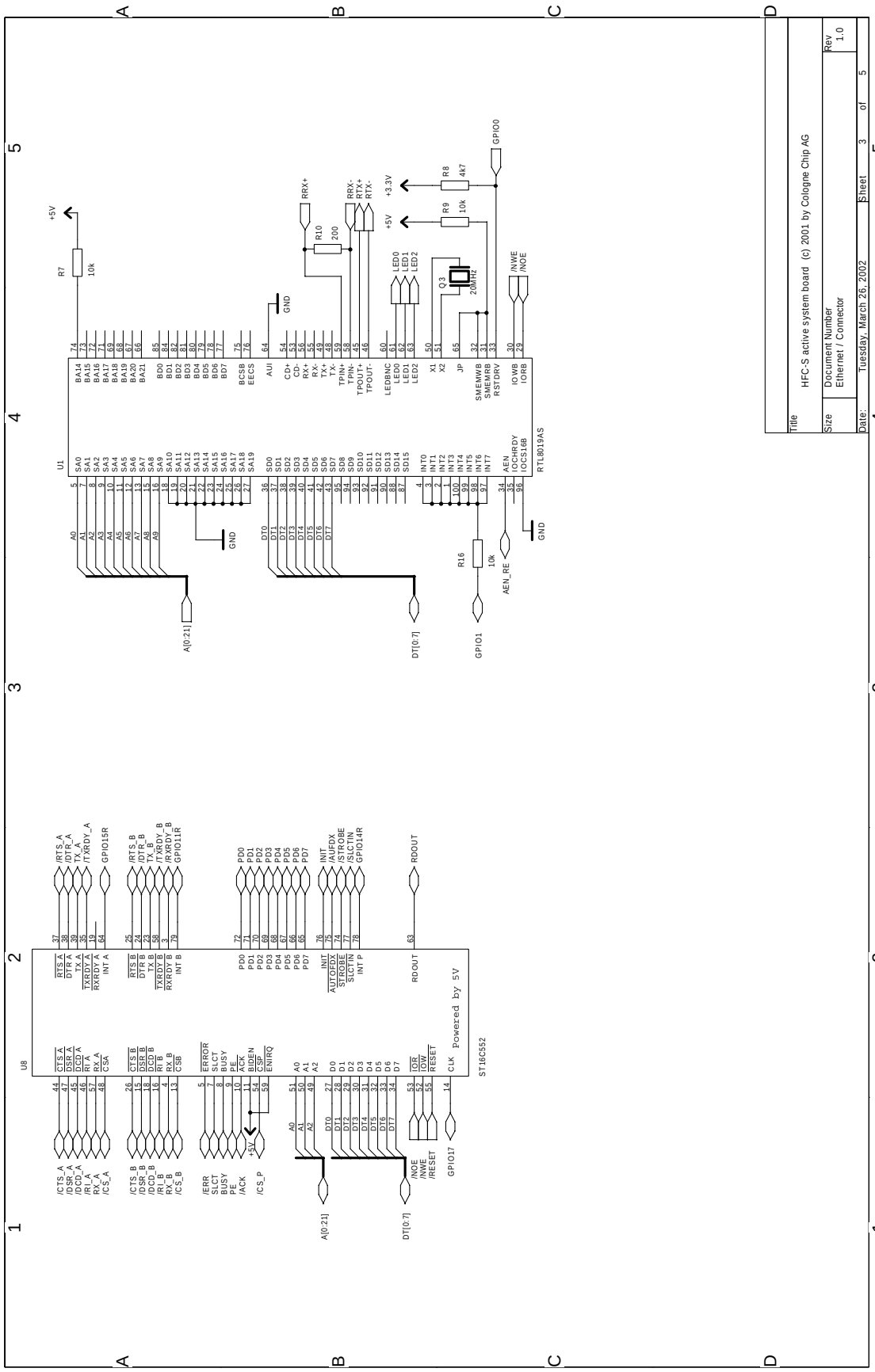

JP32
JP33
JP34
JP35
JP36
JP37
GND
+5V
+3.3V
Title
HFC-S active Eval Board (c) 2001 by CC AG
Size A4
Document Number
Power
Rev 1.0
Date: Friday, March 08, 2002
Sheet 4 of 6
U5 LM1117
C25 100n
C24 10µ
C23 10µ
C22 100n
C21 470µ
L2 220µ
R35 3k6
R36 1k2
D10 ZD
C19 470p
U4 MC34063A
R37 0.33R
C20 100µ
C18 100n
C17 100n
C16 100n
VCC
Vin
L3
L4
JP16
JP17
POW2X1
D17 gn
D16 gn
R68 1k5
R67 680

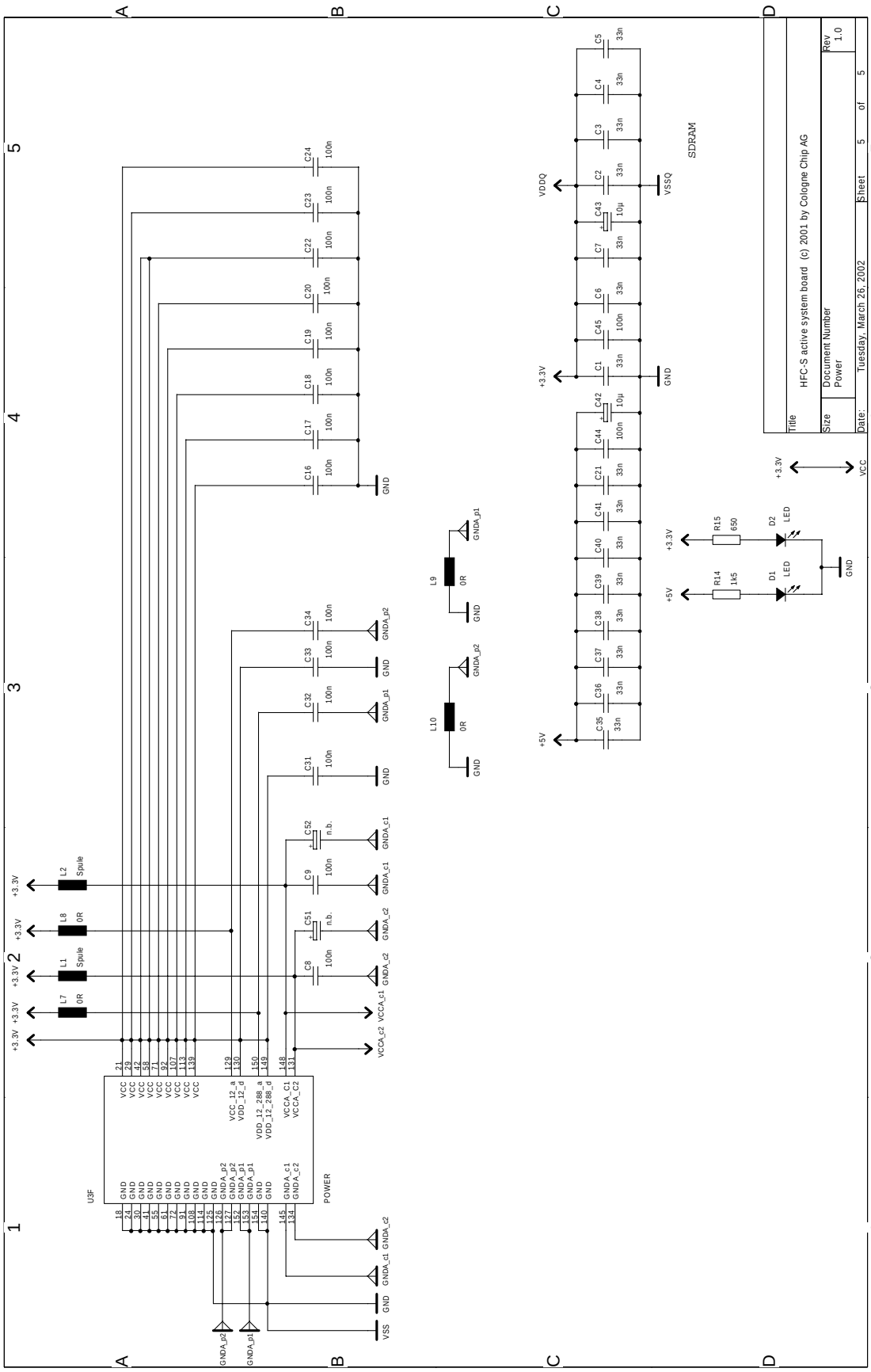

HFC-S active Eval Board (c) 2001 by CC AG
Title
Size A4
Document Number
Connector
Rev 1.0
Date: Friday, March 08, 2002
Sheet 5 of 6
sub_d_25
sub_d_9_fe
sub_d_9_ma
USB_B
MAX232
MAX1485
20F001N
RJ45
/AUFDX
/ERR
/INIT
/SLCTIN
/STROBE
PD0
PD1
PD2
PD3
PD4
PD5
PD6
PD7
/ACK
BUSY
PE
SLCT
DPUSB
DNUSB
LED1
LED0
LED2
RxD1
TxD1
CTS_1
RTS_1
RxD2
TxD2
CTS2
RTS2
CTS3
RxD3
RTS3
TxD3
RTX+
RTX-
RRX+
RRX-
H/F
GND
+5V
+3.3V

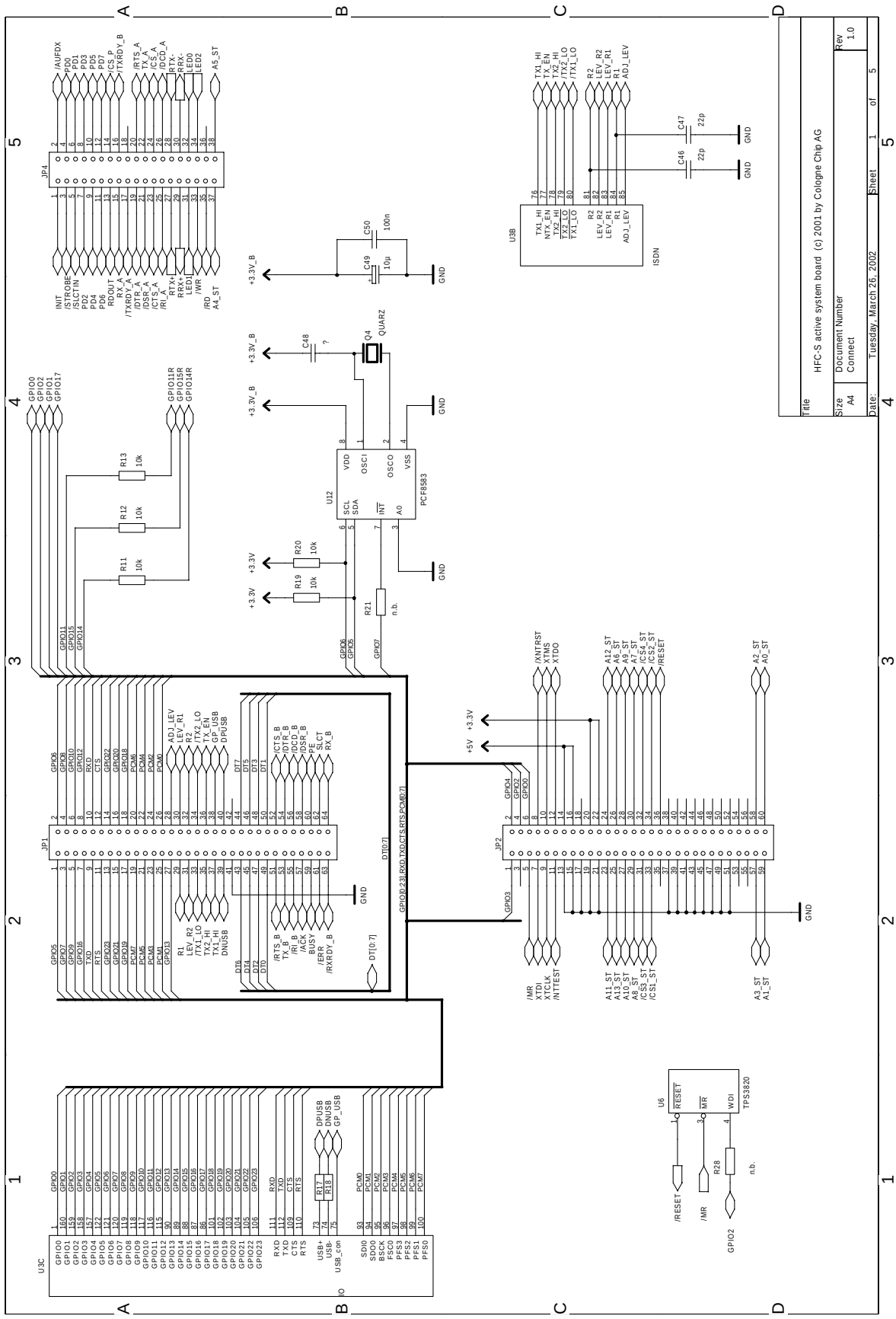

HFC-S active Eval Board (c) 2001 by CC AG
Document Number
Connector 2
Rev 1.0
Sheet 3 of 6
Wednesday, February 27, 2002
JP3
JP5
CON1X4
JP4
CON1X4
D11
LED_SMD
D12
LED_SMD
GPIO9
GPIO10
R38
750
R39
750
JP28
Taster
JP15
CON2X15
JP2
JP1
R69
0R
JP29
DIP4
R66
10k
R65
10k
JP18
CON2X10
JP21
CON2X5
JP31
CON2X10
JP30
CON2X1
JP19
JP20


Title
HFC-S active system board (c) 2001 by Cologne Chip AG
Size
A4
Document Number
Memory
Rev
1.0
Date:
Wednesday, January 16, 2002
Sheet
2
of
5
U3A
Memory
U4
29W160/320
U5
K4S641632D
U7
74LVT245
U10
74HC541
U11
74HC541
U9
74HCT138
R2
R3
R4
R5
R6
22R
+3.3V
/RESET
DT[0:7]
A[0:21]
D[0:15]
/NCS1
/NCS2
/NCS3
/NCS4
/NCS5
/NWE
/NOE
DQML
DQMU
/NRAS
/NCAS
/NWE_SD
CKE_SD
CLK_SD


Title
HFC-S active system board (c) 2001 by Cologne Chip AG
Size
Document Number
Codec / PLL
Rev
1.0
Date: Tuesday, March 26, 2002
Sheet 4 of 5
U3E
XTALOUT12
XTALIN12
FIL_PLL12
XTALOUT12_288
XTALIN12_288
FIL_PLL12_288
NRES
XNTRST
XTDI
XTMS
XTCLK
XTDO
NTESTMODE
Control
/RESET
/XNTRST
XTDI
XTMS
XTCLK
XTDO
/NTTEST
Q2
12.288MHz
R22
100
C29
22p
C28
22p
C27
22p
C26
1n
C25
1n
GND
GNDA_p1
GNDA_p2
JP7
CON1X4
JP6
CON1X4
VCCA_c1
VCCA_c2
GNDA_c1
GNDA_c2
Cod1_out
Cod1_in
Cod2_out
Cod2_in
L5
L6
Spule
L4
L3
U3D
CODEC
AINFB1
AMODIN1
VREFOUT1
APOSTOUT1
REFH1
REFL1
AINFB2
AMODIN2
VREFOUT2
APOSTOUT2
REFH2
REFL2
R23
12k
R27
10R
R25
12k
R26
10R
R1
39k
R24
39k
C53
C57
C54
C58
C60
C59
C11
C15
C56
C12
C55
C14


Title
HFC-S active system board (c) 2001 by Cologne Chip AG
Size
Document Number
Ethernet / Connector
Rev
1.0
Date:
Tuesday, March 26, 2002
Sheet
3
of
5
RTL8019AS
ST16C552
Powered by 5V
R7
10k
R8
4k7
R9
10k
R10
200
R16
10k
Q3
20MHz
+5V
+3.3V
GND
GPIO0
GPIO1
GPIO17
A[0:21]
DT[0:7]
AEN_RE
/NWE
/NOE
/RESET
/IOR
/IOW
LED0
LED1
LED2
RRX+
RRX-
RTX+
RTX-


SDRAM
POWER
U3F
HFC-S active system board (c) 2001 by Cologne Chip AG
Tuesday, March 26, 2002
Document Number
Power
Rev 1.0
Sheet 5 of 5


Title
HFC-S active system board (c) 2001 by Cologne Chip AG
Size
A4
Document Number
Connect
Rev
1.0
Date:
Tuesday, March 26, 2002
Sheet
1
of
5
JP4
JP1
JP2
U3C
U12
PCF8583
U3B
ISDN
U6
TPS3820
Q4
QUARZ
C46
C47
C48
C49
C50
R11
R12
R13
R19
R20
R21
R28
GND
+3.3V
+5V