SAF784x

One chip CD audio device with integrated MP3/WMA decoder

Rev. 02 — 9 May 2008 Product data sheet

1. General description

The SAF784x is a single-chip solution CD audio decoder with on-chip MP3 and WMA decoding, digital servo, audio DAC, sample-rate converter, preamplifier, laser driver and integrated ARM7TDMI-S microprocessor. The device contains all of the required ROM and RAM, including an internal re-programmable Flash ROM, and is targeted at low-cost compressed audio CD applications. The design is a one-chip CD audio decoder IC, with additions to allow low-cost system implementation of MP3 and WMA decoding.

2. Features

2.1 Features

- Channel decoder and digital servo
- 32-bit embedded ARM7 RISC microprocessor supporting both 32-bit and 16-bit ('Thumb') instruction sets
- Maximum ARM operating frequency of 76 MHz, equivalent to 68 MIPS
- Decoding of compressed audio stream (MP3/WMA) on ARM7 core
- All memories required for MP3/WMA decoding embedded on chip: combination of 130 kB mask-programmed internal program ROM (to reduce wait-states on high-speed code, e.g. decompression algorithms), 42 kB boot ROM, 64 kB of internal re-programmable Flash ROM (for simple re-programming of application code) 110 kB internal SRAM
- Programmable clock frequency for ARM microprocessor - allowing users to trade-off power consumption and processing power depending on requirements
- Block decoder hardware to perform C3 error correction
- Sample-rate converter circuit to convert compressed audio sample rates (in the range 8 kHz to 48 kHz) to an output rate of 44.1 kHz
- Microprocessor access to digital representations of the diode input signals from the optical pickup; the microprocessor can also generate the servo output signals RA, FO, SL, allowing the possibility of additional servo algorithms in software
- Programmable PDM outputs (effectively sine and cosine) to allow use of stepper motor for sledge mechanism
- Microprocessor access to audio streams, both from the internal CD decoder and an external stereo auxiliary input (e.g. an analog source from a tuner, converted to digital via on-chip ADCs) to allow audio processing algorithms in the ARM microprocessor, e.g. bass boost, volume control
- Four general-purpose analog inputs (A_IN1 to A_IN4) allowing the ARM microprocessor access to other external analog signals, e.g. low-cost keypad, temperature sensor, via on-chip ADCs

- Two additional analog audio inputs (AUX_L, AUX_R) to allow the ARM microprocessor access to external audio signals (e.g. tuner); allows audio algorithms (e.g. bass boost) to be performed on external audio signals
- Real-time clock operated from separate 32 kHz crystal; allows low-power Standby mode with real-time clock still operational
- Watchdog timer
- I^2S-bus, S/PDIF, subcode (V4) and subcode sync outputs
- 32 GPIOs
- Two standard UART channels
- Two external interrupt pins
- I^2C-bus interface configurable for master or slave modes, supporting 100 kbit/s and 400 kbit/s standards
- Slave I^2S-bus mode, in which the channel decoder can synchronize the CD playback speed to an I^2S-bus clock input
- Integrated digital HF/Mirror detector with measurement of minimum and maximum peak values, amplitude and offset
- Integrated CD-text decoder
- Up to 6× decode speed, CLV or CAV modes
- LQFP144 package with 0.5 mm pin pitch
- Separate left and right channel digital silence detection available on KILL pins
- Digital silence detection available on loopback data from external source as well as internal data
- ‘Filterless’ pseudo bit stream audio DAC with minimal external components
- Stereo line outputs for audio DAC
- Loopback mode allowing the use of integrated DAC with external I^2S-bus/EIAJ sources
- Compatible with voltage mode mechanisms
- On-chip buffering and filtering of the diode signals from the mechanism in order to optimize the signals for the decoder and servo parts
- LF (servo) signals converted to digital representations by Sigma-Delta ADCs shared between pairs of channels to minimize DC offset between channels
- HF part summed from signals D1 to D4 and converted to digital signals by HF 6-bit ADC
- Selectable DC offset cancellation of quiescent mechanism voltages and dark currents, digitally controlled; additional fine DC-offset cancellation in digital domain
- Eye pattern monitor system to observe selectable points within the analog pre-amp
- Current and average jitter values available via registers
- On-chip laser power control, up to maximum currents of 120 mA
- Laser on-off control, including ‘soft’-start control - zero-to-nominal output power in 1 ms
- Monitor control and feedback circuit to maintain nominal output power throughout laser life
- Configured for Nsub (N-substrate) monitor diode
- JTAG interface for device access and ARM code development (compatible with ARM multi-ICE)
- All digital input pins 5 V tolerant
- Low-latency static memory interface to access a maximum of two 2 MB memory

- This product has been qualified in accordance with AEC-Q100

2.2 Formats

Reads the following CD-decode formats

- CD-R
- CD-RW
- CD-DA (*CD Red Book; IEC 60908*)
- CD-ROM (Mode 1 and Mode 2)
- CD-MP3
- CD-WMA
- Video CD
- SACD (CD layer only)
- Support 80 minute to 100 minute CD playback
- Multi-session discs

3. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
SAF7846HL	LQFP144	plastic low profile quad flat package; 144 leads; body 20 × 20 × 1.4 mm	SOT486-1
SAF7847HL			

4. Block diagram

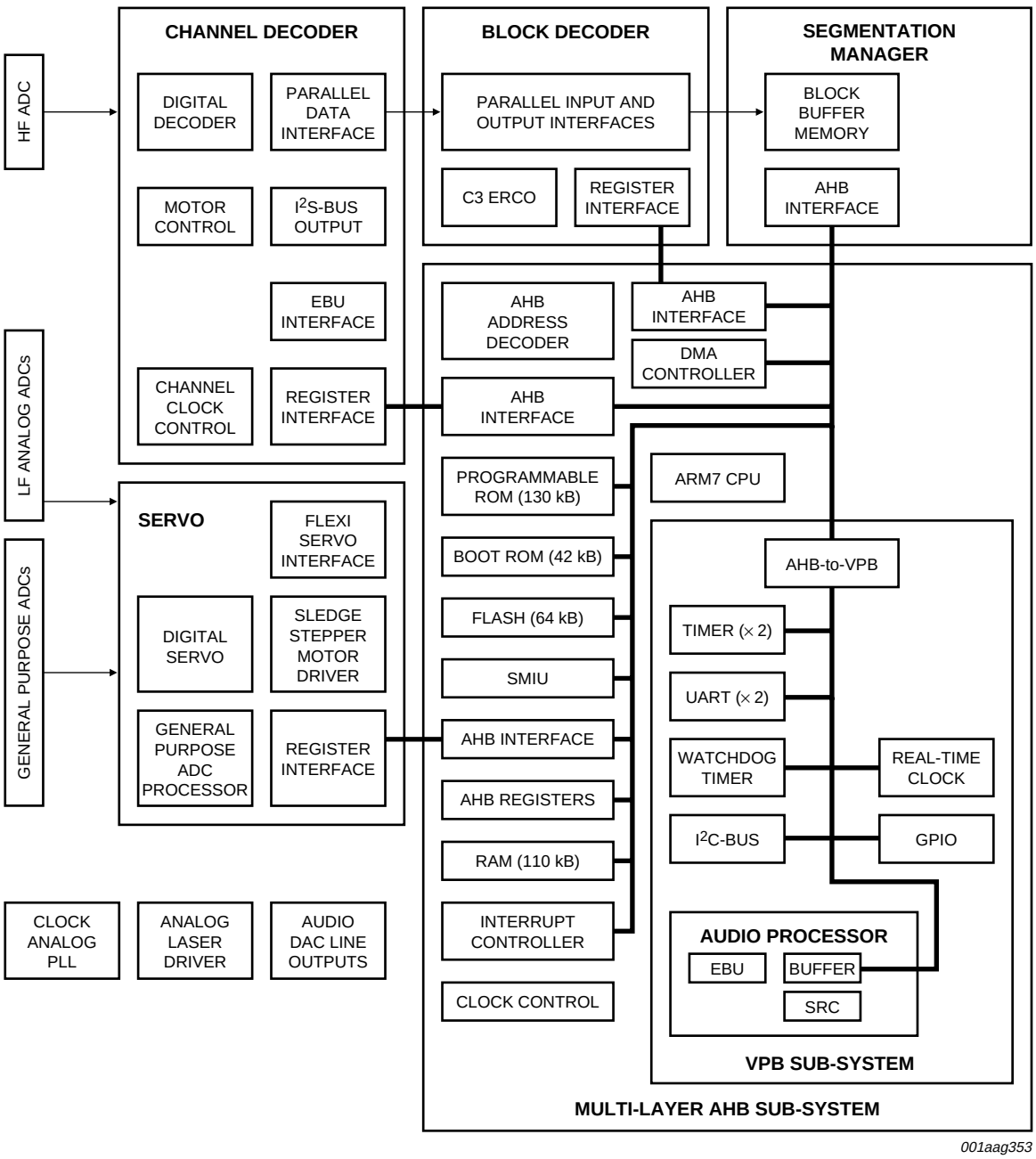


Fig 1. SAF784x top level block diagram

5. Pinning information

5.1 Pinning

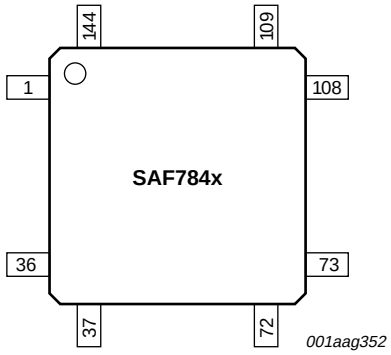


Fig 2. SAF784x pinning diagram

5.2 Pin description

Table 2. Pin description

All digital inputs and bidirectional pins are 5 V tolerant.

Symbol	Pin	Type[1]	Description
SL_SIN	1	O	sledge actuator/stepper motor PDM output (sine)
COS/GPIO31	2	B	stepper motor PDM output (cosine)/general purpose I/O 31
LPOWER	3	P	laser power supply
LASER	4	P	laser diode drive
MONITOR	5	AI	laser monitor diode input
VSSA1	6	P	analog ground 1
HF_MON	7	AIO	HF monitor output signal
VDDA1	8	P	analog supply voltage 1
D1	9	AI	central diode signal voltage input
D2	10	AI	central diode signal voltage input
D3	11	AI	central diode signal voltage input
D4	12	AI	central diode signal voltage input
R1	13	AI	satellite diode signal voltage input
R2	14	AI	satellite diode signal voltage input
AUX_L	15	AI	auxiliary audio left signal input
AUX_R	16	AI	auxiliary audio right signal input
VDDA2	17	P	analog supply voltage 2
OPU_REF_OUT	18	AO	OPU reference voltage
VSSA2	19	P	analog ground 2
OSCOUT	20	AO	crystal or resonator output
OSCIN	21	AI	crystal or resonator input
VDDA3	22	P	analog supply voltage 3

Table 2. Pin description *...continued*

All digital inputs and bidirectional pins are 5 V tolerant.

Symbol	Pin	Type[1]	Description
DAC_LP	23	AO	audio DAC left channel differential output (positive)
DAC_LN	24	AO	audio DAC left channel differential output (negative)
DAC_VREF	25	AIO	audio DAC decoupling point (10 µF/100 nF to ground)
DAC_RN	26	AO	audio DAC right channel differential output (negative)
DAC_RP	27	AO	audio DAC right channel differential output (positive)
DAC_FGND	28	P	audio DAC floating ground
VSSA3	29	P	analog ground 3
OSC_32K_IN	30	AO	32 kHz crystal input
OSC_32K_OUT	31	AO	32 kHz crystal output
VDDD1	32	P	digital core supply voltage 1
A_IN1/GPIO0	33	AIB	analog input 1/general purpose I/O 0
A_IN2/GPIO1	34	AIB	analog input 2/general purpose I/O 1
A_IN3/GPIO2	35	AIB	analog input 3/general purpose I/O 2
A_IN4/GPIO3	36	AIB	analog input 4/general purpose I/O 3
VSSD1	37	P	digital core ground 1
TX/GPIO4	38	B	UART transmit/general purpose I/O 4
RX/GPIO5	39	B	UART receive/general purpose I/O 5
TX2/GPIO6	40	B	UART2 transmit/general purpose I/O 6
DM_ADDR_0	41	O	external memory address bit 0
RX2/GPIO7	42	B	UART2 receive/general purpose I/O 7
DM_ADDR_1	43	O	external memory address bit 1
SDA	44	B	micro interface data I/O line (open-drain output)
DM_ADDR_2	45	O	external memory address bit 2
SCL	46	B	microprocessor interface clock line
DM_ADDR_3	47	O	external memory address bit 3
LKILL	48	O	kill output for left channel (configurable as open-drain)
DM_ADDR_4	49	O	external memory address bit 4
RKILL	50	O	kill output for right channel (configurable as open-drain)
VSSP1	51	P	digital ground 1 to periphery (pads)
DOBM	52	O	bi-phase mark output (no external buffer required)
VDDP1	53	P	digital supply voltage 1 to periphery (pads)
DM_ADDR_5	54	O	external memory address bit 5
INT2/GPIO8	55	B	external interrupt 2/general purpose I/O 8
DM_ADDR_6	56	O	external memory address bit 6
GPIO9	57	B	general purpose I/O 9
DM_ADDR_7	58	O	external memory address bit 7
GPIO10	59	B	general purpose I/O 10
DM_ADDR_8	60	O	external memory address bit 8
GPIO11	61	B	general purpose I/O 11
DM_ADDR_9	62	O	external memory address bit 9

Table 2. Pin description *...continued*

All digital inputs and bidirectional pins are 5 V tolerant.

Symbol	Pin	Type[1]	Description
GPIO12	63	B	general purpose I/O 12
DM_ADDR_10	64	O	external memory address bit 10
GPIO13	65	B	general purpose I/O 13
DM_ADDR_11	66	O	external memory address bit 11
GPIO14	67	B	general purpose I/O 14
DM_ADDR_12	68	O	external memory address bit 12
GPIO15	69	B	general purpose I/O 15
DM_ADDR_13	70	O	external memory address bit 13
SDI/GPIO16	71	B	serial data input (loopback)/general purpose I/O 16
DM_ADDR_14	72	O	external memory address bit 14
WLCI/GPIO17	73	B	serial word clock input (loopback)/general purpose I/O 17
DM_ADDR_15	74	O	external memory address bit 15
SCLI/GPIO18	75	B	serial bit clock input (loopback)/general purpose I/O 18
VSSD2	76	P	digital core ground 2
VDDD2	77	P	digital core supply voltage 2
DM_ADDR_16	78	O	external memory address bit 16
T1/GPIO19	79	B	tacho input 1 (for spindle motor sensor)/general purpose I/O 19
DM_ADDR_17	80	O	external memory address bit 17
T2/GPIO20	81	B	tacho input 2 (for spindle motor sensor)/general purpose I/O 20
DM_ADDR_18	82	O	external memory address bit 18
T3/GPIO21	83	B	tacho input 3 (for spindle motor sensor)/general purpose I/O 21
DM_ADDR_19	84	O	external memory address bit 19
PWM1/CAP1/GPIO22	85	B	timer PWM output 1/capture input 1/general purpose I/O 22
DM_ADDR_20	86	O	external memory address bit 20
PWM2/CAP2/GPIO23	87	B	timer PWM output 2/capture input 2/general purpose I/O 23
DM_BLS_0	88	O	external RAM lower-byte lane select (lower 8-bits)
PWM3/CAP3/GPIO24	89	B	timer PWM output 3/capture input 3/general purpose I/O 24
DM_BLS_1	90	O	external RAM upper byte lane select (upper 8-bits)
PWM4/CAP4/GPIO25	91	B	timer PWM output 4/capture input 4/general purpose I/O 25
DM_WE	92	O	external memory right control
MEAS/GPIO26	93	B	channel decoder telemetry output/general purpose I/O 26
DM_OE	94	O	external memory output enable
CFLG/GPIO27	95	B	channel decoder correction statistics/general purpose I/O 27
DM_CE_0	96	O	external memory chip-select Bank 0
CL1/GPIO28	97	B	clock output for sampling channel decoder telemetry outputs/general purpose I/O 28
GPIO29	98	B	general purpose I/O 29
VSSP2	99	P	digital ground 2 to periphery (pads)
$\overline{\text{RESET}}$	100	IUH	power-on reset (active LOW)
VDDP2	101	P	digital supply voltage 2 to periphery (pads)

Table 2. Pin description *...continued*

All digital inputs and bidirectional pins are 5 V tolerant.

Symbol	Pin	Type[1]	Description
DM_CE_1	102	O	external memory chip-select Bank 1
INT1	103	IUH	external interrupt 1
DM_DATA_0	104	B	external memory data input/output bit 0
VSSD3	105	P	digital core ground 3
VDDD3	106	P	digital core supply voltage 3
EF	107	O	C2 error flag
DM_DATA_1	108	B	external memory data input/output bit 1
DATA	109	O	serial data output
DM_DATA_2	110	B	external memory data input/output bit 2
WCLK	111	O	word clock output
DM_DATA_3	112	B	external memory data input/output bit 3
SCLK	113	O	serial clock output
DM_DATA_4	114	B	external memory data input/output bit 4
SYNC	115	O	EFM frame synchronization
DM_DATA_5	116	B	external memory data input/output bit 5
V4/CL16	117	B	versatile pin 4/clock output 16.9344 MHz
DM_DATA_6	118	B	external memory data input/output bit 6
TDI	119	IU	JTAG1/2 test data input
DM_DATA_7	120	B	external memory data input/output bit 7
TMS	121	IU	JTAG1/2 test mode select
DM_DATA_8	122	B	external memory data input/output bit 8
TCK	123	IDH	JTAG1/2 test clock
DM_DATA_9	124	B	external memory data input/output bit 9
$\overline{\text{TRST}}$	125	IU	JTAG1/2 asynchronous reset (active LOW)
DM_DATA_10	126	B	external memory data input/output bit 10
TDO	127	O	JTAG1/2 test data output
DM_DATA_11	128	B	external memory data input/output bit 11
ARM_JTAG_SEL	129	I	select ARM JTAG (active HIGH) or general JTAG (active LOW)
DM_DATA_12	130	B	external memory data input/output bit 12
RTCK/GPIO30	131	B	JTAG clock output/general purpose I/O 30
DM_DATA_13	132	B	external memory data input/output bit 13
DEV_ROM	133	ID	development ROM select (LOW = internal ROM)
DM_DATA_14	134	B	external memory data input/output bit 14
VSSD4	135	P	digital core ground 4
VDDD4	136	P	digital core supply 4
DM_DATA_15	137	B	external memory data input/output bit 15
MOTO1	138	O	motor output 1
MOTO2	139	O	motor output 2
VSSP3	140	P	digital ground 3 to periphery (pads)
VDDP3	141	P	digital supply voltage 3 to periphery (pads)

Table 2. Pin description *...continued*

All digital inputs and bidirectional pins are 5 V tolerant.

Symbol	Pin	Type[1]	Description
RA	142	O	radial actuator
FO	143	O	focus actuator
n.c.	144	-	not connected pad

[1] See Table 3 for pin type definition.

Table 3. Pin type definition

Type	Definition
AI	analog input
AO	analog output
AIO	analog input/output
AIB	analog input or bidirectional
ID	digital input with pull-down
IDH	digital input with pull-down and hysteresis
IU	digital input with pull-up
IUH	digital input with pull-up and hysteresis
O	digital output, slew-rate limited
B	digital bidirectional, slew-rate limited
P	power connection

6. Functional description

6.1 Analog data acquisition

The input signals from the OPU photodiodes contain information used in the servo loops and the high frequency data from which the audio samples are reconstructed. The SAF784x contains all the necessary circuitry to process the photodiode signals directly and hence removes the need for a separate external diode signal preamplifier.

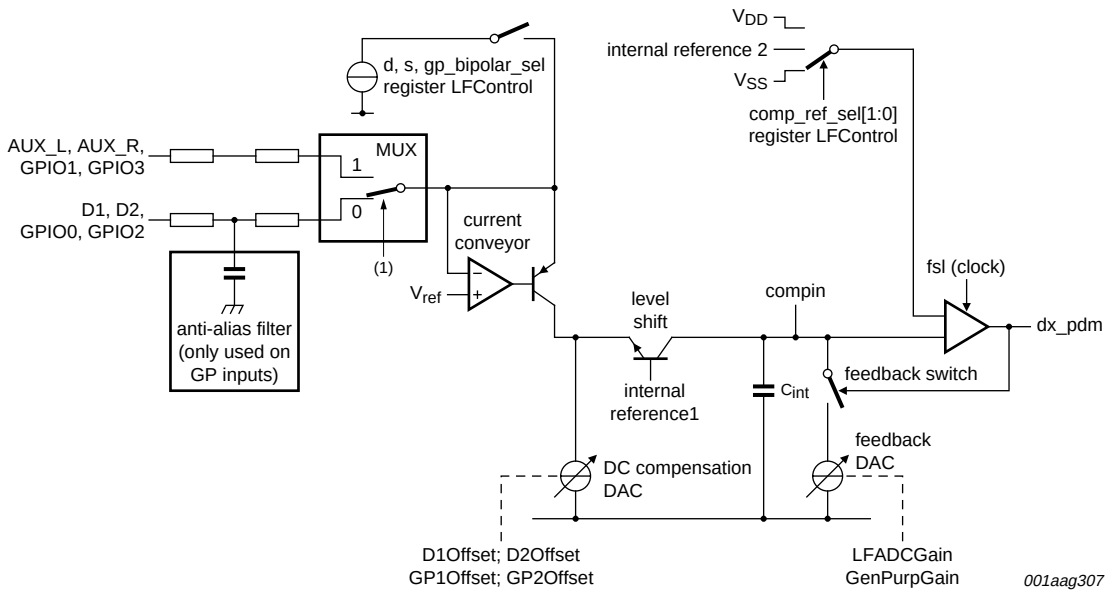
6.1.1 LF acquisition

The LF signal path acquires the photodiode voltage signals and converts them into 4 MHz pulse-density modulated digital data streams. These streams are processed within the digital servo to control the focus, radial and sledge loops.

The servo processing makes use of the difference calculations D1 – D2, D3 – D4 and R1 – R2. Ideally these differences should be zero when the quantities D1 to R2 are equal due to the laser illumination. However in a practical system, errors reduce the accuracy of the signal processing. Two main forms of errors exist - DC offsets and relative gain mismatch between the difference channels.

The DC offsets are minimized in the SAF784x by DC-offset compensation circuitry which allows the DC present in the Pulse Density Modulation (PDM) streams to be measured when the laser is switched off, and then subtracted from the signals in the digital domain when the laser is on.

Relative gain mismatch is minimized by using carefully scaled circuitry in the time-continuous parts of the signal path, and by time-sharing circuitry in the time-discrete parts. A simplified block diagram of the LF acquisition path is shown in Figure 3.



(1) if_auxin_sel = 0: D1 and D2 selected; if_auxin_sel = 1: AUX_L, AUX_R selected.
if_gpio_sel = 0: GPIO0 and GPIO2 selected; if_gpio_sel = 1: GPIO1 and GPIO3 selected.

Fig 3. LF acquisition block diagram

The output of the OPU is converted to a current across the input resistor. The current conveyor provides a low input impedance and a high output impedance and sets a virtual earth at the end of the voltage-to-current converter to the same voltage as V_{ref} (1.6 V).

The level shifter acts as a summing node for the DC cancellation and produces a current that is referenced to an internal bias voltage which is independent of V_{ref}.

The output current charges an integration capacitor C_{int}. When the voltage reaches V_{DDA} / 2, the comparator switches and sends a feedback current that has opposite polarity to the input current which tries to discharge the capacitor.

The register LFADCGain defines the amount of feedback current and so sets the ADC gain. A PDM waveform appears at the output of the ADC, and is passed through a low-pass filter (in the digital domain). The average value at the output of the filter is in proportion to the voltage between V_i and V_{ref}.

Input signals from the OPU, GPIO inputs and the AUX inputs are routed to eight ADCs comprising six LF ADCs and two general purpose ADCs. ADCs LF1, LF2, GP1 and GP2 are shared by some of these inputs which are routed via an internal multiplexer. ADC LF1 is shared by input pairs D1, D2, and AUX_L, AUX_R via the multiplexer. ADCs LF3 to LF6 are dedicated to inputs D3, D4, R1 and R2 respectively. ADC GP2 is shared by input pairs GPIO0, GPIO2, and GPIO1, GPIO3 via the multiplexer. The internal multiplexer is controlled by register AuxandGPADCControl.

6.1.2 HF acquisition

The HF data (EFM) signal is obtained by summing the signals from the three or four central diodes of the OPU and filtering and converting the signals to a digital representation via a 6-bit HF ADC. Figure 4 shows a simplified block diagram of the HF path.

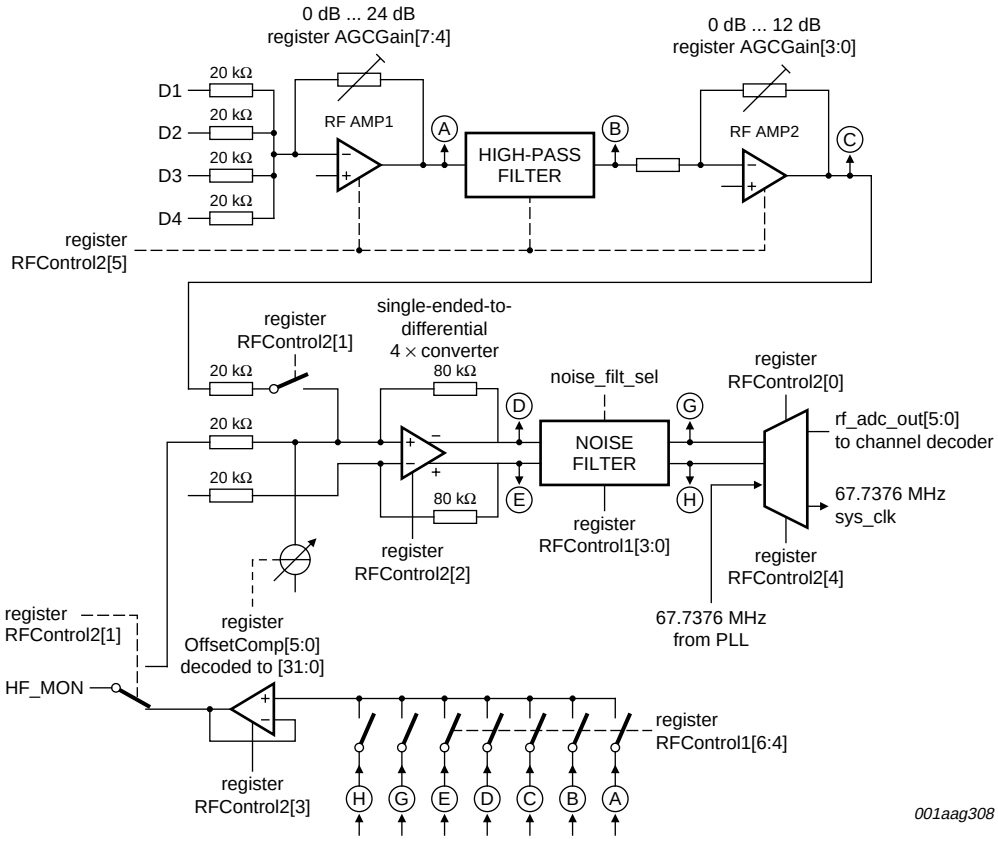


Fig 4. HF acquisition block diagram

The four diode signals D1, D2, D3 and D4 are summed in the first RF amplifier. The gain of the first amplifier is controlled by register AGCGain[7:4].

A second gain stage has been added to lessen the gain bandwidth requirements of a single gain stage operational amplifier and also to act as dynamic Automatic Gain Control (AGC). The gain of this amplifier is set by register AGCGain[3:0] and can be changed on-the-fly from the ARM microprocessor. The gain range accommodates 12 dB of gain needed to boost the signal when the laser tracks across a fingerprint defect on the disc.

CD-R, CD-RW, and finger prints, not only reduce the AC signal amplitude compared to a perfectly pressed disc but also reduces the DC pedestal voltage. The high-pass filter will remove all DC present at the input, but offsets will be added by the second and third gain stages. A 5-bit plus-sign DAC controlled by register OffsetComp[5:0] adds a current to compensate for this offset. The current reduces in linear dB steps and follows the AC gain.

To help users set up the correct gain and DC offset for each particular mechanism, an eye pattern monitor facility is included. This consists of a high frequency buffer amplifier whose input can be selected to monitor various important nodes within the analog RF path. The monitor point is controlled by register RFControl1[6:4] field RFMONSEL. The output of the buffer drives HF_MON pin (pin 7). This register also controls the roll-off frequency of the noise filter which is in front of the 6-bit ADC in the RF path.

Various blocks within the analog RF path can be powered down if required, including the complete path. These power-down bits are controlled by register RFControl2[5:0].

In addition, the 6-bit RF ADC can be stand-alone tested in application mode, or a separate external RF path IC can be connected to SAA7834 by selecting bit 1 of register RFBypassSel. The input for the RF signal is then via pin HF_MON. In this mode the central diode summing circuit, RF AMP1, high-pass filter and RF AMP2 are all bypassed.

6.2 Analog clock generation

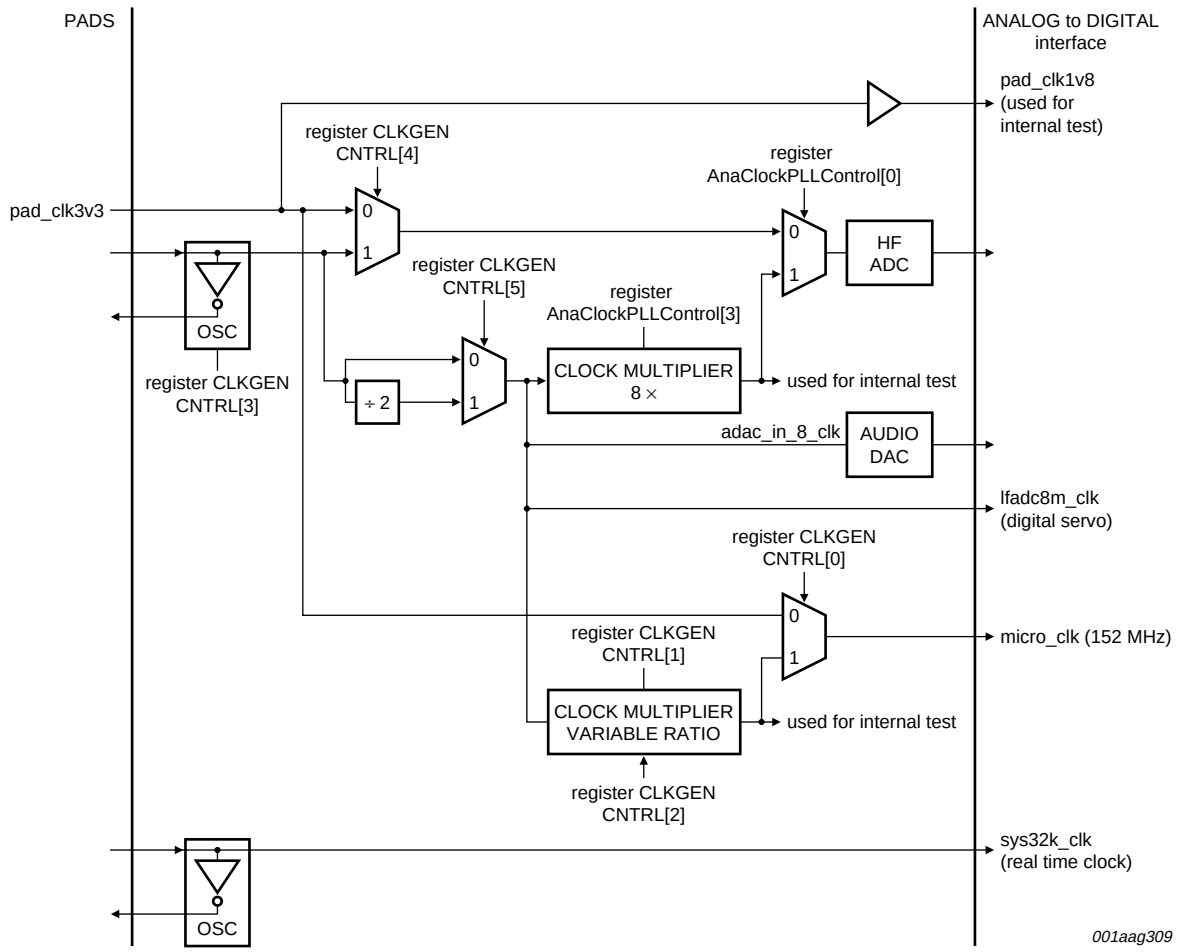


Fig 5. Analog clock generation

The SAF784x consists of two analog phase-locked loops. The 67 MHz PLL is dedicated to the channel decoder. The 152 MHz PLL is dedicated to the remaining functionality. The clock strategy for the SAF784x is intended to address areas that are prone to noise effects

that can decrease the quality of audio. The clocks related to audio DAC and LF ADC are generated directly from the analog signal, instead of being derived from high frequency PLLs. The clocking strategy for the digital core is shown in Figure 9.

6.3 General purpose analog inputs

The four general purpose ADC inputs, GPIO0 (pin 33), GPIO1 (pin 34), GPIO2 (pin 35), and GPIO3 (pin 36), can be used for giving the ARM microprocessor access to external analog sources, such as monitoring temperature and to provide simple resistor-ladder keypad functionality. These inputs use an additional pair of sigma-delta ADCs identical to those used for the LF diode inputs.

The general purpose analog inputs have separate interrupt request lines and use address space in the servo registers for storing the converted digital values. The output of the general purpose ADCs are low-pass filtered and can have fine-offset compensation added before being passed to a decimation filter. The digital values output from the decimation filter are then captured in the servo registers with a resolution of 10 bits per channel.

There are only two ADCs for general purpose application and so each ADC is multiplexed between two inputs: ADC1 between GPIO0 and GPIO2, and ADC2 between GPIO1 and GPIO3. GPIO2 and GPIO3 inputs are selected by signal AuxControlandGPADC.

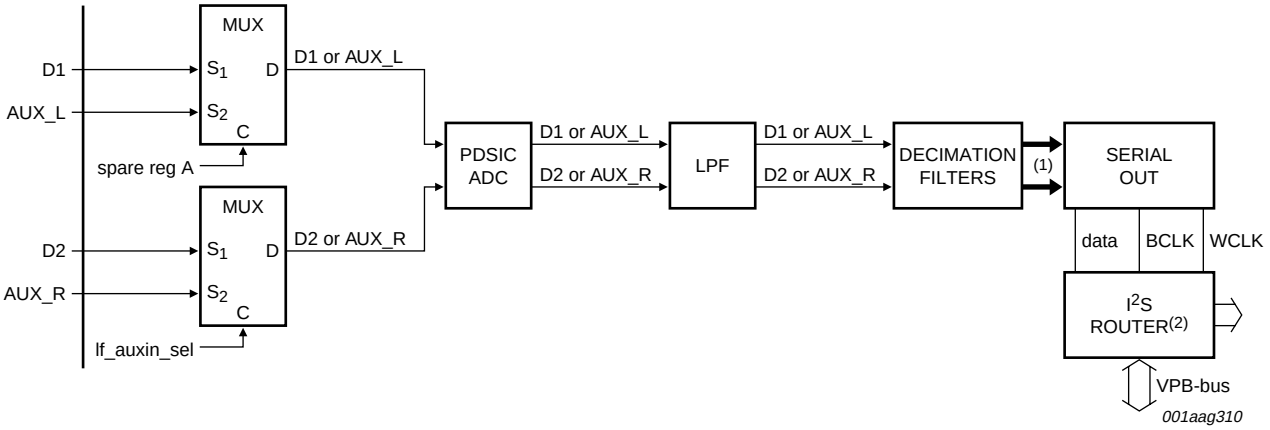
6.4 Auxiliary analog inputs

Analog inputs, AUX_L and AUX_R, are available, and have sufficient resolution, for the input of external audio sources, such as allowing ARM access to an external audio source for sound processing algorithms. This allows audio processing of external audio sources via the AUX pins, whilst simultaneously using the general purpose inputs for keyboard and temperature inputs.

Since these two inputs share one pair of the LF sigma-delta ADCs used in the LF path (for inputs D1 and D2) a multiplexer is used to control the data source into the ADCs. Therefore, D1 and D2 cannot be used at the same time as AUX_L and AUX_R. This path is designed for a tuner input where the THD specification is ~0.3 % and the S/N is < 60 dB. These performance values are lower than when the normal CD audio path is used i.e. S/N > 80 dB and THD < 0.01 %.

The audio data is converted to a pulse-density modulated digital stream for both input channels. This data is then low-pass filtered and decimated to produce 10-bit representations of the analog inputs.

The auxiliary inputs differ from the general purpose analog inputs because the parallel data is converted to an I^2S format stream and then sent to the I^2S handler block to make the data available to the ARM microprocessor. The I^2S handler contains a 12-deep data FIFO which allows the ARM microprocessor to service the audio data with a lower priority than it would need if it were directly registered; see Figure 6.



(1) 10-bit samples of AUX_L and AUX_R expanded to 16 bits by adding 6 LSBs.

(2) I^2S router block used to route I^2S to audio DAC or pins.

Fig 6. Auxiliary analog inputs block diagram

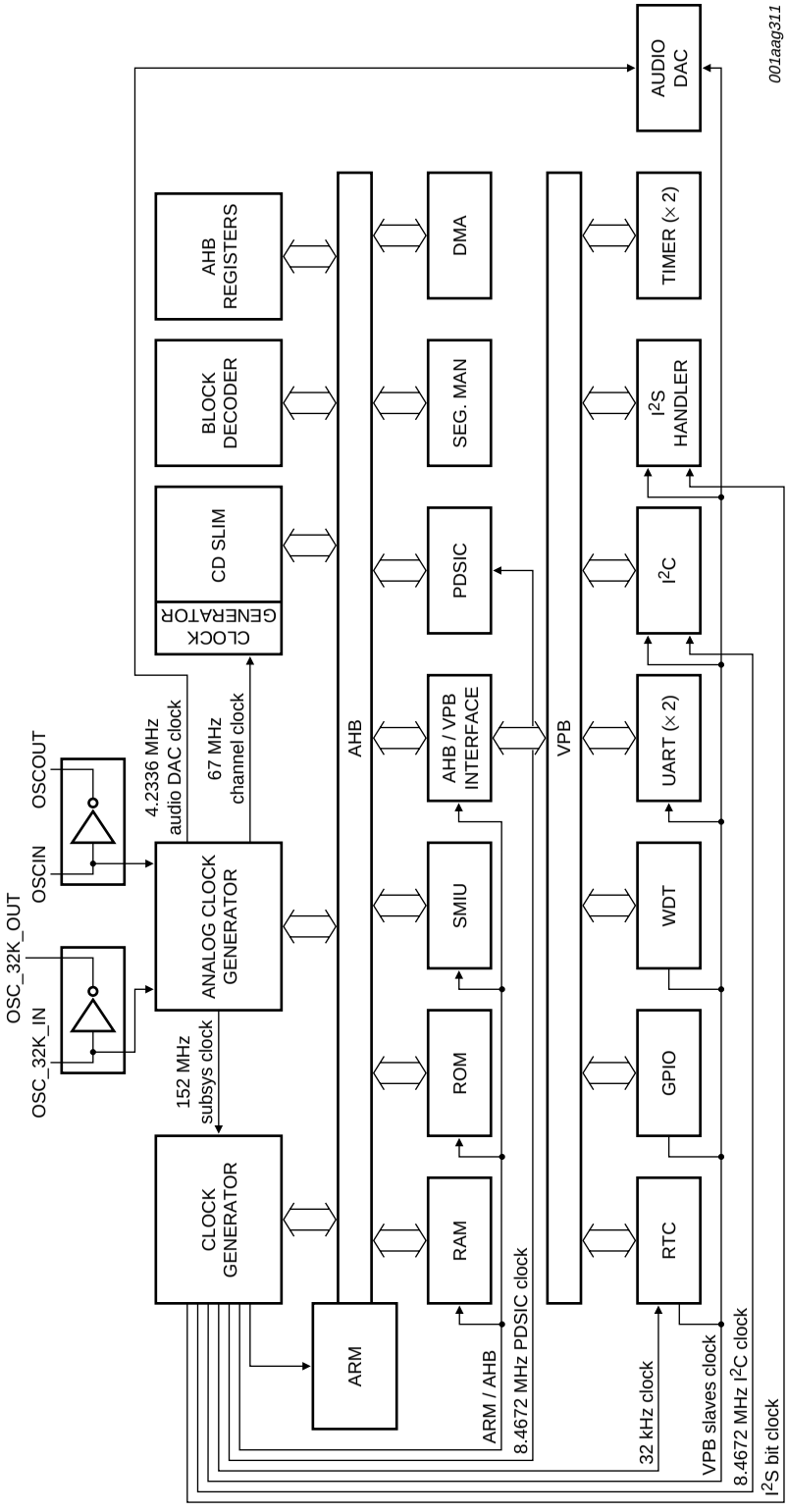


Fig 7. Clocking top-level block diagram

6.5 Channel decoder

6.5.1 Features

The channel decoder in the SAF784x is derived from the design used in the SAA7817HL DVD decoder IC. The design has been optimized for CD decode functionality (EFM and demodulation is removed) and has the following features:

- One-channel interface to the on-chip 6-bit 67 MHz ADC
- Signal conditioning logic with high-pass filter, DC-offset cancellation (AOC) and AGC logic
- HF defect detection circuitry with automatic hold of AGC, AOC, HPF, PLL and slicer on defect detection
- Digital equalizer, noise filter, PLL and slicer
- Run Length 2 (RL2) push back mechanism
- EFM demodulator with sync interpolation
- CD text and subcode Q-channel extraction blocks with software interface via registers
- Decoding, de-interleaving and Reed-Solomon error correction according to CD CIRC standards
- On-chip de-interleaving SRAM memory
- Audio processing back end with interpolate/hold, mute, kill and silence detect logic, de-emphasis and 4× upsample filter
- Two data output interfaces: I^2S and EBU
- One serial subcode output interface (V4)
- Motor control for CLV (locked-on EFM) or CAV (locked-on tacho) or open loop or software-controlled regulation with one or two motor pins
- On-board tacho measurement with one or three Hall sensor inputs (T1 to T3), that provides frequency input for motor loop; the sensor inputs are shared with GPIO pins
- 8-bit register map, with AHB slave interface
- An interrupt output with associated interrupt, status and interrupt enable registers for full interrupt-driven operation
- Debug information available via Meas1 (CL1, pin 97) and pin CFLG (pin 95) and parallel debug bus

6.5.2 Block diagram

The incoming diode signals are first added and processed in the analog front end to create a normal RF (HF) signal and then converted to digital by the ADC. This signal is then resampled from the ADC clock to the system clock domain via the integrate-and-dump block. Offset and gain on the RF signal is removed via the AGC/AOC loop (via the analog front end). Any remaining offset which is not removed by the analog front end can be removed via the digital HPF. The RF signal is then sliced by the bit detector, clock recovery is done by a full digital PLL with noise filter, equalizer and sample-rate convertor. A defect detector allows AGC, AOC, HPF, slicer and PLL to be held during black or white dots. At this point in the data path, RF samples are converted into a bit stream. The RL2 push back avoids RL3s in the RF being accidently translated into RL1 or RL2 in the bit stream. The channel bit stream is demodulated in to bytes by the EFM demodulator. Q-channel subcode and CD-text information is extracted via the Q-subcode

and CD-text decoder, available for readout through the sub-CPU interface. The main data stream is error-corrected by the ERCO, while the memory processor takes care of the CIRC de-interleaving and buffering of data in a FIFO. At the back end of the channel decoder, corrupted audio samples can be interpolated and held, while a burst of errors can trigger the mute block. Detection of digital silence can be used to kill the internal or external audio DAC. Pre-emphasis on the audio disc can be removed via the de-emphasis filter, and the data can be 4× upsampled before it is sent to the audio DAC. CD data is output via the I^2S and/or the EBU outputs. Motor control can be frequency-regulated to the incoming RF bit rate, with additional phase regulation by FIFO filling, or it can be fully controlled via software. This method guarantees CLV support. A tacho measurement block is also available. The motor can also be regulated by the tacho frequency which allows possible CAV support.

Debug information is available via registers, via the dedicated serial lines Meas1 and Cflg.

The ARM AHB address of registers that control specific logic are shown next to each functional block in Figure 8.

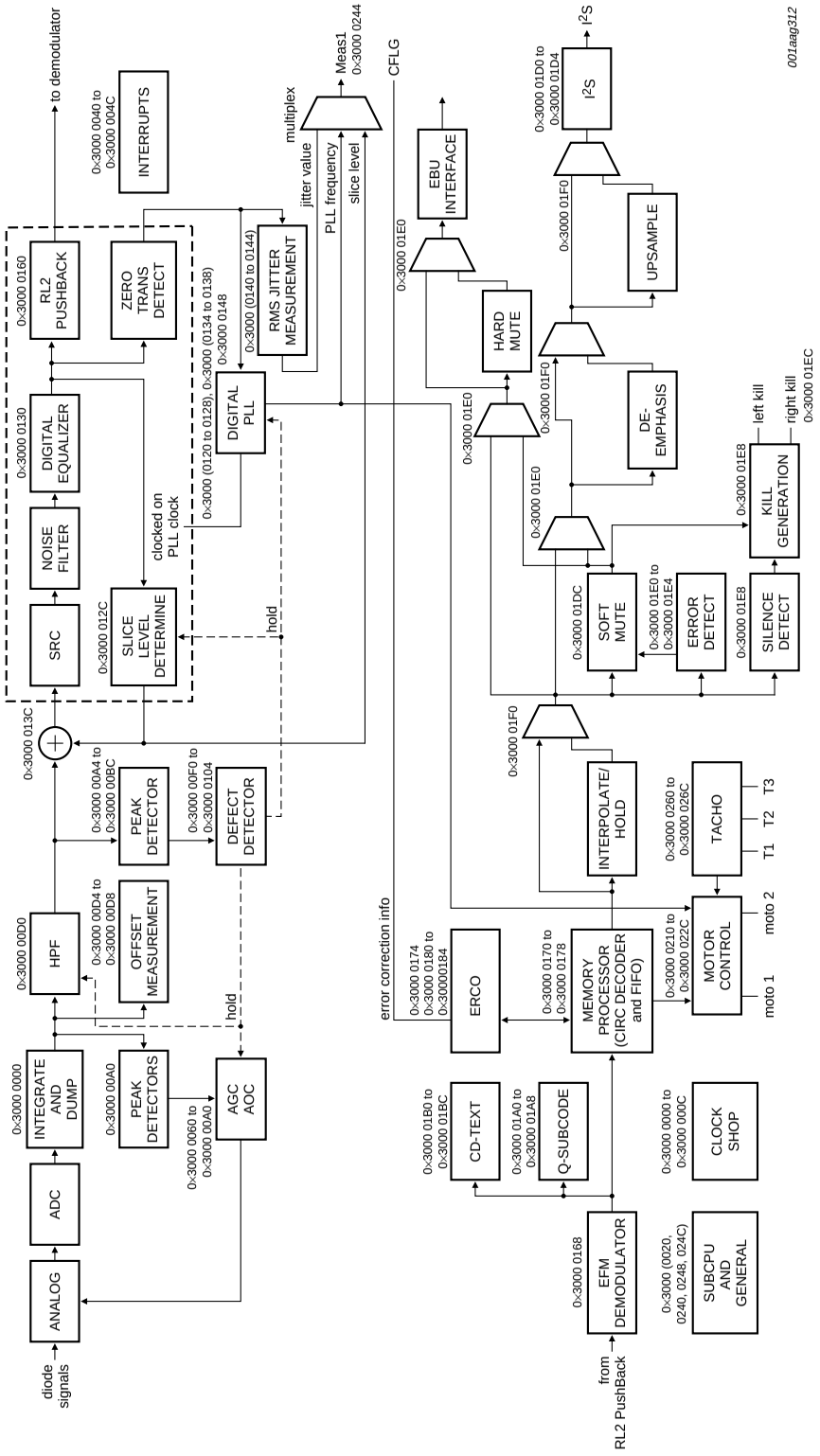
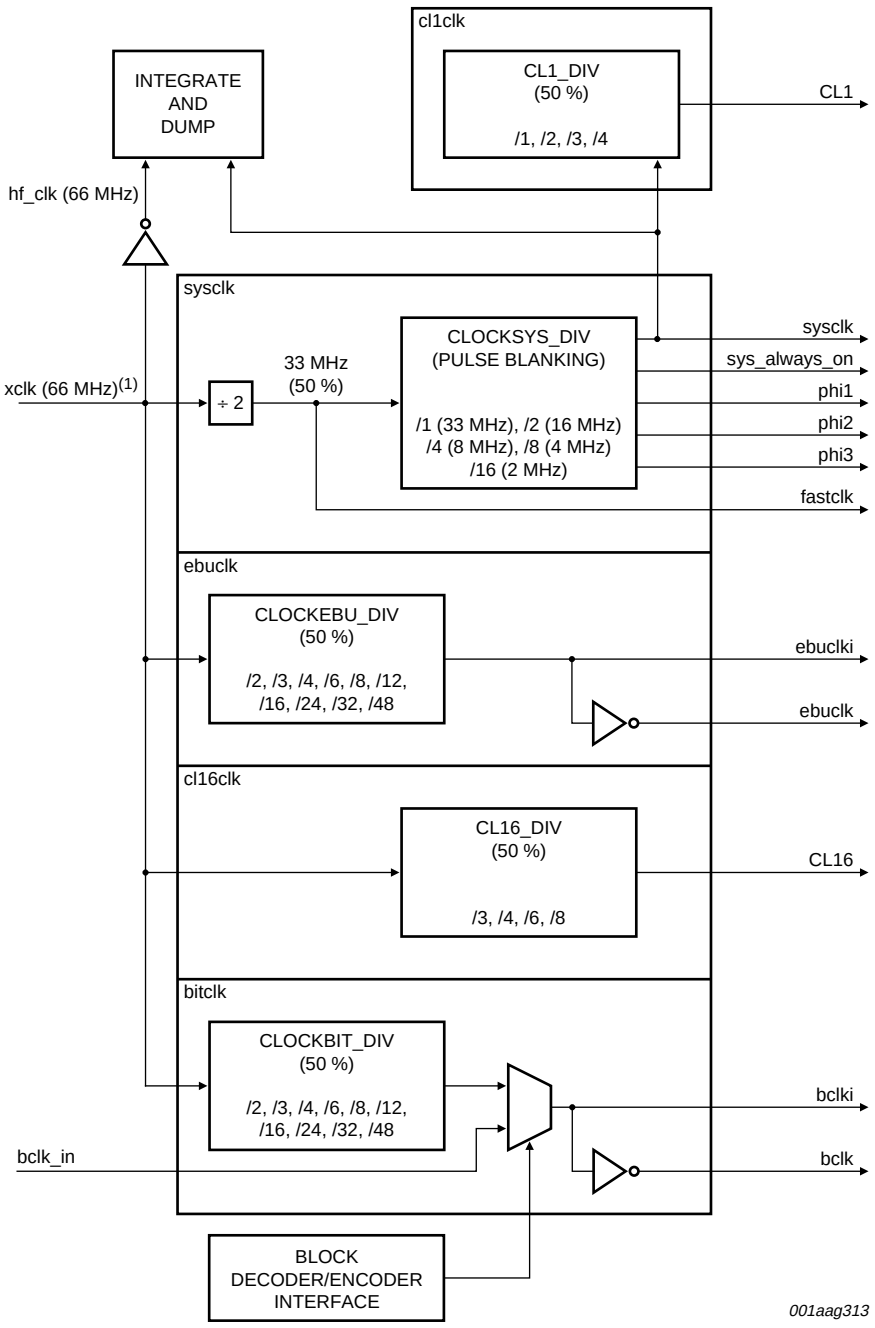


Fig 8. Channel decoder top-level block diagram

6.5.3 Clock control



(1) 66 MHz is an approximate value.

Fig 9. Clock control block diagram

The clock control block defines the clock frequencies for four clock domains.

- xclk: most internal clocks are derived from the crystal clock. This clock is the output of the clock multiplier in the analog part and has a fixed frequency of 67.7376 MHz = 8.4672 MHz (f_{xtal}) × 8. If a 16 MHz crystal is used, the crystal clock is divided by 2 inside the analog block. Crystal selection is done via AnalPLLControl bit SEL16.
- sysclk domain: the system clock, or its derivatives, runs the main part of the internal channel decoder. The sysclk is derived from xclk divided by 2 (50 % duty cycle) and can be further divided down via register SysclockConfig bit SYSDIV. This register also allows the majority of clocks to be powered down (for Sleep mode). The choice of the sysclk frequency $f_{clk(sys)}$ in an application is determined by the expected input bit rate f_{bit} of the RF bit stream. The relationship between this incoming bit stream frequency and the system clock frequency $f_{clk(sys)}$ is expressed by the ratio $f_{bit} / f_{clk(sys)}$. There are two limiting factors:
  - The HF-PLL operating range is between $0.25 \times (f_{bit} / f_{clk(sys)})$ and $2 \times (f_{bit} / f_{clk(sys)})$
  - The decoder and error corrector throughput rate is limited to $1.7 \times (f_{bit} / f_{clk(sys)})$

  This brings the constraint to $0.25 < f_{bit} / f_{clk(sys)} < 1.7$.
- bitclk domain: runs the I²S back-end logic. The bit clock (bitclk) is also output as part of the I²S interface. In audio slave mode this clock must be programmed to be exactly 44100 Hz × 2 × 16/24/32 (depending on I²S mode), to get a 1× data rate to the audio DAC. In master mode with gated bitclk, the bitclk must be programmed to be at a higher rate than the outgoing bit rate required for the disc speed, to avoid FIFO overflow in the decoder. For example, at N = 1, the incoming RF bit rate = 4.3218 MHz, which corresponds to an output bit rate of 1.4112 MHz. This means that the bitclk frequency is above 1.4112 MHz and is high enough when I²S-16 is chosen, while I²S-32 requires the bitclk to be at least 2.8224 MHz. The bitclk division is selected via register BitClockConfig. Also, bitclk gating can be enabled via the same register.
- ebuclk domain: runs the EBU back end. The EBU (or S/PDIF) interface is only enabled during audio slave mode. The ebuclk needs to be exactly 44100 Hz × 64 = 2.8224 MHz for 1× operation. The ebuclk division is selected via register EBUClockConfig.

  The following clocks are also controlled by the clock control block:
  - The hf_clk is fixed at 67.7376 MHz, and is used to clock-in the samples from the ADC, which is clocked by the xclk with the same clock frequency
  - The bclk_in is the incoming I²S bit clock, which is used when I²S is programmed to receive bclk rather than transmitting it (programmed via register I2SConfig)
  - The cl1clk can be used to monitor the Cflg and Meas1 debug lines. The frequency can be programmed via register CLClockConfig
  - The cl16clk can be used to clock an external audio DAC or audio filter IC. The frequency can be programmed via register CLClockConfig

6.5.4 Decoder-ARM microprocessor interface

The decoder core is internally connected to the ARM core via the AHB interface for register access to the decoder internal configuration registers.

6.5.4.1 Programming interface

Decoder registers are programmed through the AHB interface. The programming interface is not fully described in this document.

For the application, it should be noted that the interface supports 32-bit registers, while the decoder only contains 8-bit registers. Therefore, the decoder registers are treated as 32-bit registers of which the 24 MSBs are not used.

The register address map occupied by the decoder goes from relative address 0x3000 0000 to address 0x3000 0374, and can be split into two parts:

0x3000 0000 - 0x3000 024C: the decoder's own registers - used to configure the channel decoder, and the functionality they control is described in detail in this section.

0x3000 02A0 - 0x3000 0374: the decoder immigrant registers - used to control parts of the SAF784x that do not have their own AHB interface (they are not used to control the decoder channel decoder).

6.5.4.2 Interrupt strategy

The channel decoder contains two interrupt status registers: InterruptStatus1 contains all interrupts that operate as set/reset latches (set by hardware, reset by reading from the register). InterruptStatus2 contains all interrupts that operate as feed-throughs (set by hardware, reset by hardware or by accessing other registers).

Each interrupt bit can be enabled or disabled separately by writing to its corresponding enable bit in the InterruptEnable1 and InterruptEnable2 registers. If one or several interrupt bits are set and at least one is enabled, the interrupt line of the decoder to the microcontroller will go active (LOW). If an interrupt bit is disabled (enable bit turned off), it is prevented from activating the interrupt line to the microcontroller. However, this mode allows the interrupt to be processed if the status register is polled instead of interrupt handling by the microcontroller.

6.5.5 EFM bit detection and demodulation

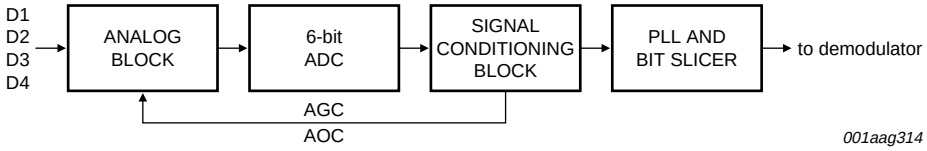


Fig 10. Bit recovery block diagram

A block diagram of the bit recovery is shown in Figure 10.

The HF signal comprises the four diode inputs inside the analog block. It is pre-processed (LPF, HPF, offset removal and gain adjustment) and then sampled by a 6-bit ADC.

On the sampled HF, bit recovery is done by means of a full digital PLL and slicer.

Before the sampled signal enters the PLL section, it is pre-processed by a signal conditioning block. This consists of an integrate-and-dump block, a high-pass filter and logic available for gain control and offset control on the RF signal in the analog section.

For good playability on defects, a defect detector puts the PLL, the slicer, the AGC, the offset cancellation and the high-pass filter into hold during defects.

The detected bits are then sent to the demodulator for sync extraction and EFM demodulation. For playing on damaged or out-of-specification discs, flywheels are in place to make the sync extraction more robust.

6.5.5.1 Signal conditioning

This device has a number of blocks which process the incoming 6-bit HF signal.

- Integrate-and-dump block to adapt the frequency of the ADC to the system clock
- Peak detection logic for amplitude measurement
- Peak detection logic for DC-offset measurement
- Digital high-pass filter with configurable cut-off frequency
- DC and gain control logic for on-board variable gain and offset control (in the analog section)
- A defect detector

All blocks can be configured under microprocessor control.

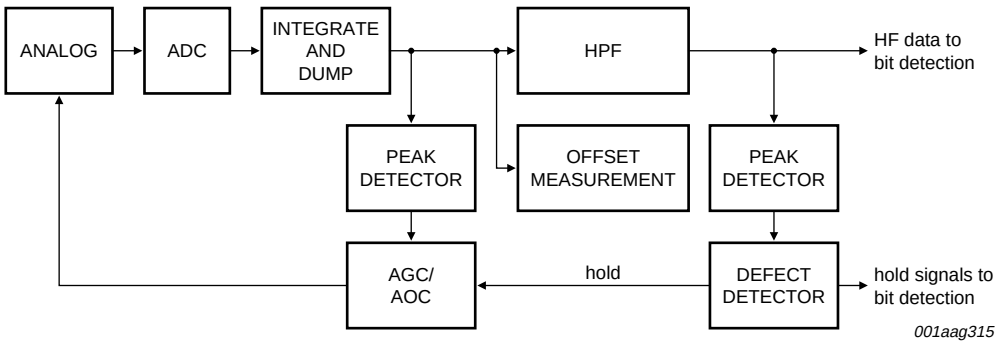


Fig 11. Signal conditioning block diagram

Integrate-and-dump block: the ADC delivers one sample every xclk period (= one sample every hf_clk period). The sample rate needs to be adapted from this xclk rate to the lower sysclk rate. For more information on sysclk speed, see Section 6.5.3 on page 19. The integrate-and-dump block converts the incoming samples at the hf_clk frequency into a stream of one sample per sysclk period. It converts an average of a number of samples to achieve this. If the division factor for the system clock is /2, /4, /8, /16, /32, an average of 2, 4, 8, 16 or 32 incoming samples respectively, is taken and passed further. This results in a gain in the number of effective bits of the A-D conversion.

High-pass filter: A 1st-order IIR high-pass filter with a variable 3 dB point is implemented to filter out the remaining DC jump-on defects. Most of these defects will have been filtered by the analog HPF. The cut-off frequency of the digital high-pass filter can be changed on-the-fly, by writing to register HighPassFiltCont.

It is possible to reset the state of the high-pass filter via bit 6 of register HighPassFiltCont. The input and the output of the high-pass filter are 8-bits wide.

The high-pass filter is implemented in a ‘1 minus low pass’ structure. It is possible to hold the low-pass filter on defects. For more information, see Section “Defect detector” on page 26.

The high-pass filter works on the system clock. Its bandwidth is also proportional to the sysclk.

A formula for approximating the cut-off frequency (f_c), of the high-pass filter is:

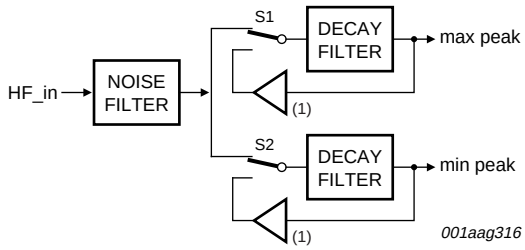
$$f_{c,HPF} = \frac{HPSet[5:0]}{2\pi \times 2^{11}} \times f_{clk(sys)} \quad (1)$$

Peak Detectors: The signal conditioning block has two types of peak detector:

- peak detector with decay filter: works on an immediate attack/slow-decay basis, and is used for measuring peaks, amplitude and offset, read by software which sends peak information to the defect detector.
- peak detector based on window: works on the principle of detecting maximum and minimum peaks within a window, and is used for the AGC and AOC control logic.

Both peak detectors monitor the RF after it has passed an optional noise filter. This noise filter is a LPF with a programmable high cut-off frequency. This bandwidth is programmed via register PDBandwidth bit NOISEFILTERBW for the noise filter before the peak detectors of AGC/AOC and measurement read back. The defect detector peak detector has its own noise filter which is programmed via register DefectDetPeakBW bit NOISEFILTERBW.

Peak detector with decay filter: The functional schematic of this peak detector is shown in Figure 12.



(1) Determines the time-constant coefficient.

Fig 12. Peak detection block with decay filter

The maximum and minimum peaks of the incoming signal are measured at the inputs of switches S1 and S2 respectively. The maximum and minimum peak signal paths both have a decay filter with a long time-constant and matching bandwidth. The maximum peak decay filter responds to the smallest value possible. The decay filter for the minimum peak responds to the largest value possible.

The decay bandwidth of the measurement readback decay filter is controlled by register PDBandwidth bit DECAYBW, the bandwidth of the defect detector is controlled via register DefectDetPeakBW bit DECAYBW.

The following settings of the decay filters are possible: $C = 1 - 2^{-m}$, for m = 6 to 21, where

C = time-constant coefficient, m = DecayBW[3:0] + 6.

The bandwidths and corresponding time-constant (t) of the decay filter are shown in Table 4, when the system clock frequency $f_{clk(sys)}$ is 10 MHz.

Table 4. Decay filter time-constants at $f_{clk(sys)}$ = 10 MHz

m	t	m	t	m	t	m	t
6	6.35 µs	10	102.4 µs	14	1.64 ms	18	26.21 ms
7	12.75 µs	11	204.7 µs	15	3.28 ms	19	52.43 ms
8	25.55 µs	12	409.6 µs	16	6.55 ms	20	104.8 ms
9	51.15 µs	13	819.2 µs	17	13.11 ms	21	209.7 ms

Peak detector based on window: The functional schematic of this peak detection is shown in Figure 13.

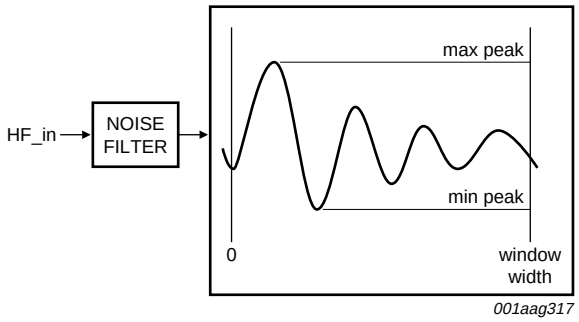


Fig 13. Peak detection block with window

The minimum and maximum peaks of the incoming signal are measured during a programmable window period. The highest and lowest sample within this window are used to update maximum and minimum peaks.

The window width of the measurement is controlled via register AGCAOCControl bit PDMEASWINDOW.

AGC and AOC control block: The AGC control block controls the RF amplitude at the input of the ADC by controlling the gain of an on-chip analog gain amplifier. The AOC control block controls the RF offset at the input of the ADC by adding or subtracting the offset just before the ADC. Both AGC and AOC loops are built up in the same manner and are shown in Figure 14 in their relative position within the signal conditioning block.

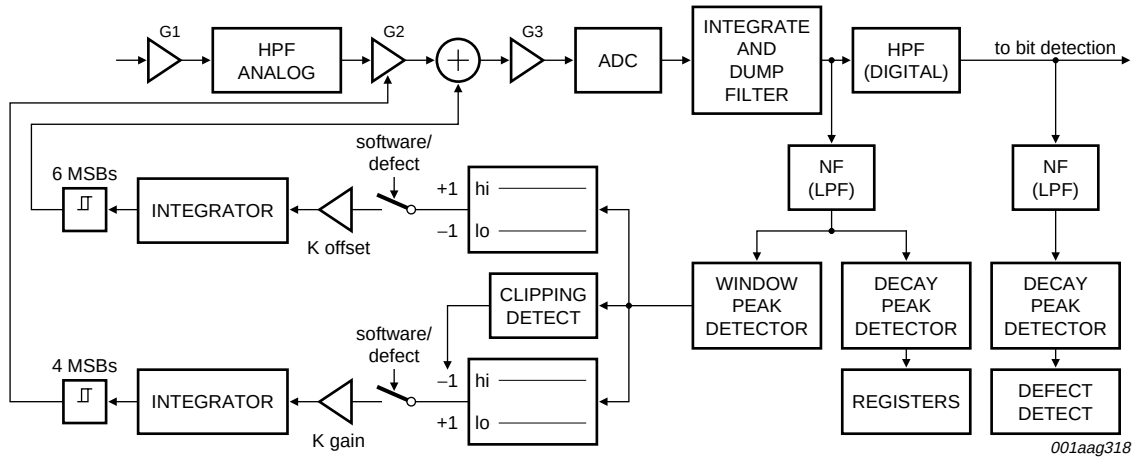


Fig 14. AGC and AOC loops

The maximum and minimum peaks on the envelope of the RF signal after the ADC are first measured via a noise filter and the window peak detector (see Section "Peak Detectors" on page 23. The amplitude is then calculated as maxpeak – minpeak, and the offset as (maxpeak + minpeak) / 2.

For tuning the loops, it is possible to read back the HFMaxPeak, HFMinPeak, HFAmplitude and HFOffset, as measured from their registers by the decay peak detector.

AGC control: The RF amplitude at the ADC input can be changed with two gain amplifiers in the analog part: G1 (fixed) and G2 (dynamic). G1 has a gain range from 0 dB to 24 dB in 16 steps of 1.6 dB, while G2 has a range from 0 dB to 12 dB in 16 steps of 0.8 dB. Both gains can be programmed via register AGCGain. G1 will stay fixed, while G2 can be regulated by hardware when the AGC is turned on.

The AGC will regulate the gain such that the measured amplitude stays between a programmed upper threshold (AGCThrHi) and lower threshold (AGCThrLo). If the amplitude is smaller, gain will increase; if the amplitude is too large, gain will decrease. When clipping is detected on either one or both sides, the gain will decrease. These gain changes are not sent to the analog gain amplifier directly but are integrated over time. Only if, on average, a gain increase or decrease is requested, will this result in a real gain increase or decrease of the amplifier. This can also be read back via register AGCGain. The AGC, together with the noise filter on the peak detector, prevents RF noise causing over-sensitive gain regulation. To further reduce sensitive behavior, a hysteresis window with a width of one gain step is added between the integrator and amplifier G2. The bandwidth of the gain loop determines how fast it reacts to fingerprints and scratches; it is programmed via register AGCIntegBW. It is also possible to limit the range of G2 by programming a maximum and minimum boundary by register AGCGainBound.

AOC control: Most of the RF offset at the ADC input will be removed by the analog HPF (1st-order HPF with 3 dB point around 3.6 kHz). The remaining offset (mainly introduced by the analog front end), can be removed by adding or subtracting a fixed offset in the analog part. This offset subtraction/addition has a range of 32 steps in each direction, with approximately 1.4 LSBs per step (referenced to the RF ADC). This leads to a full

correction range of ± 42 LSB steps (more than the whole ADC range). This offset compensation (offset comp) value can be programmed via register OffsetComp, and will be regulated in hardware as soon as the AOC is turned on.

The AOC will regulate the offset comp value such that the measured offset stays within a window programmed by register OffsetBound. The offset comp value decreases if the offset is above this window, and increases if the offset is below the window. If an inversion occurs on the RF signal between analog and digital, the reaction of this loop can be inverted by programming OffsetBound bit OFFSETINV.

The offset changes are not sent to the analog offset subtraction directly, but are integrated over time. Only if, on average, an offset increase or decrease is requested, this will result in a real offset increase or decrease of the analog addition. This can also be read back via register OffsetComp. The AOC, together with the noise filter on the peak detector, prevents RF noise causing over-sensitive offset regulation. To further reduce sensitive behavior, a hysteresis window with a width of one offset step has been added between the integrator and offset comp value. The bandwidth of the offset loop will determine how fast it reacts to fingerprints and other defects; it is programmed via register OffsetIntegBW. It is also possible to limit the range of the offset comp value by programming a maximum and minimum boundary by register OffsetCompBoundHi and OffsetCompBoundLo.

AGC/AOC general and rules-of-thumb: The AGC and AOC hardware regulation loops can be enabled or disabled separately via register AGCAOCControl. This register also allows the use of a ‘slow’ AGC and/or AOC loop. In this case the programmed loop bandwidth is decreased with an extra factor of 128. In this mode the loops will be too slow to react to defects, but can be used for a slow software-like gain and/or offset regulation to regulate the average gain and offset over the disc comfortably within a specified range.

An important feature is the AGCAOCControl bit DISHOLDNOLOCK, which disables holding of the AGC and AOC loops during defects (triggered by the defect detector, see Section “Defect detector” on page 26 while the HF PLL is not in lock. This feature avoids permanent lockups of the loops caused by a small amplitude triggering the defect detector, which in return would hold the AGC loop.

The following things should be taken into account as general ‘rules-of-thumb’:

- The amplitude thresholds should not be programmed too close to each other: allow at least two gain steps (1.6 dB) from lower to higher boundary and vice versa to avoid an over-sensitive AGC.
- The offset boundary should not be programmed too tight: ± 8 is a good value to avoid an over-sensitive AOC.

The bandwidth of the loops should never be programmed to be too wide (‘fast’) with respect to the peak detector measurement window, to avoid an unstable loop. If the PDwindow = 2^n $f_{clk(sys)}$ (Hz) wide, the bandwidth of the loops should never be higher than $2^{-(n+1)}$ (Hz).

Defect detector: The defect detector detects the presence of black or white dots in the RF stream, and freezes some signal conditioning and bit recovery logic during these defects. This prevents the control loops inside this logic drifting away from their optimal point of operation when there is no RF present, so that they can recover very fast when good RF is present again.

The detection of a defect is based on amplitude. The amplitude is measured via a set of peak detectors with decay, as described in Section “Peak Detectors” on page 23. The decay bandwidth and noise filter bandwidth are programmed by register DefectDetPeakBW.

Two thresholds can be programmed. A low threshold will trigger a ‘defect-detected’ signal as soon as amplitude goes below this threshold. A high threshold will clear this ‘defect-detected’ signal again as soon as amplitude goes above this threshold. Together, these thresholds apply a hysteresis to the defect detection, avoiding a jittery ‘defect-detected’ signal (having many on/off parts) when the amplitude is at the threshold edge. Thresholds are programmed in register DefectDetThres.

The ‘defect-detected’ signal can be used to hold the PLL, slicer, AGC, AOC and HPF during a defect. The feature(s) that will be held can be programmed in register DefectDetEnables. The same register can be used, via software, to force the PLL, slicer and HPF to hold. The AGC and AOC can be held in software by just disabling the loops in register AGCAOCControl.

Two special features exist on the defect detector:

- Optional delay of the enabling and disabling of hold features at the beginning and end of a defect. This can be done by programming a start and/or stop delay (in number of sysclks) via register DefectDetStartStopDelay. Whenever the defect detector detects the start of a defect, it will wait for the start delay before triggering a ‘defect-detected processed’ signal. When the defect detector detects the end of a defect, it will wait for the programmed stop delay before clearing the defect-detected processed’ signal again. This also means that defects which are smaller than the start delay are ignored, and that if the defect contains zones with good RF amplitude but smaller than the stop delay, they are also ignored. In reality, all hold features are triggered by the defect-detected processed’ signal, rather than the ‘defect-detected’ signal; at the decoder output, both delays are zero, so both signals are equal.
- Optional programmable time-window at the end of a defect, during which, higher PLL and/or slicer bandwidths can be used to speed up the recovery of these loops after a defect. This window can be programmed via register DefectDetHighBWDelay. Bandwidth programming is explained in Section 6.5.5.2 on page 28.

The detection of the beginning or end of a defect, with or without start and stop delays, can be used to generate an interrupt. See register InterruptEnable1.

6.5.5.2 Bit detector

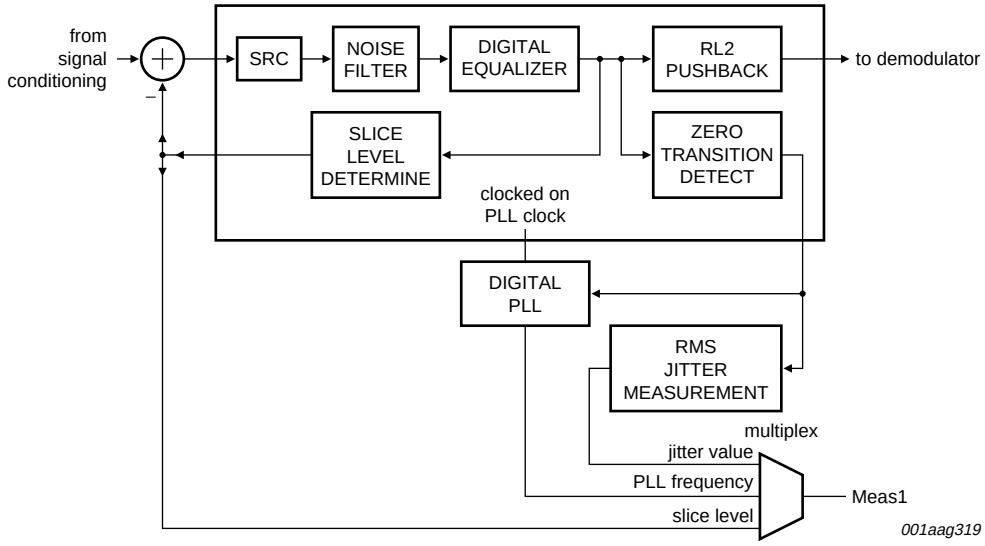


Fig 15. Bit detector block diagram

The bit detector block contains the slice-level circuitry, a noise filter to limit HF EFM signal noise contribution, an equalizer, a zero-transition detector, a run-length push-back circuit, a digital PLL and jitter measurement logic.

All processing is performed on the bit clock, and bandwidths are proportional to the channel bit rate. To achieve this, RF data is resampled from the system clock domain to the bitclk domain by making use of a sample-rate convertor. Blocks can be configured under microcontroller control and are described in detail in the next paragraphs.

Noise filter: The digital noise filter runs on the channel bit clock frequency $f_{clk(bit)ch}$. It limits the bandwidth of the incoming signal to ¼ of the channel bit clock frequency.

Passband: $0 \times f_{clk(bit)ch}$ to $0.22 \times f_{clk(bit)ch}$ (Hz)

Stop-band: $(0.28 \times f_{clk(bit)ch})$ to $(f_{clk(bit)ch} - 0.28 \times f_{clk(bit)ch})$ (Hz)

Rejection: −28 dB

Slice-level determination: The slice-level determination circuit compensates for the incoming signal asymmetry component. Bandwidth of the slice-level determination circuit is programmable via register SlicerBandwidth. Also the higher bandwidths for use after a defect (see Section “Defect detector” on page 26) are programmed in this register. The bandwidth is proportional to the channel bit clock frequency. The slice level, or asymmetry, can be read back via register SlicerAssym.

Equalizer: In the bit detection circuit, a programmable equalizer is used to boost the high frequency content of the incoming signal.

The equalizer includes an integral five-tap presentable, asymmetrical equalizer. The equalizer block diagram is given in Figure 16.

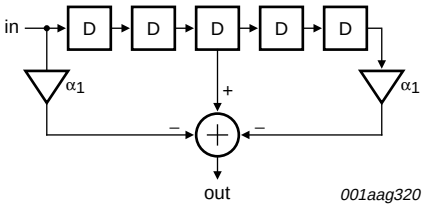


Fig 16. Equalizer block diagram

The first and last tap can be programmed via register PLLEqualiser.

Usable EFM bit clock range: The channel bit clock frequency $f_{clk(bit)ch}$ must obey the following constraints in relation to the system clock frequency $f_{clk(sys)}$. The channel bit clock frequency must always be:

- less than $f_{clk(sys)} \times 2$ (Hz)
- greater than $f_{clk(sys)} \times 0.25$ (Hz)

Therefore the range is: $f_{clk(sys)} \times 0.25 < f_{clk(bit)ch} < f_{clk(sys)} \times 2$ (Hz)

A reliable bit detection is only possible within this range. If the input channel bit rate is above $f_{clk(sys)} \times 2$ then the PLL will saturate to twice the system clock frequency.

Note that while these are theoretical limits, a real-life application should have a safety margin. When the bit clock amplitude is relatively low, the internal filter will filter off more noise, yielding a better performance. If the theoretical upper limit is approached, playability (e.g. black dot performance) will drop significantly. The decoder will only be able to correct the biggest correctable burst error of 16 frames if $f_{clk(bit)ch} < f_{clk(sys)} \times 1.7$ (Hz).

Taking this restriction on the decoder into account, the new range becomes:

$f_{clk(sys)} \times 0.25 < f_{clk(bit)ch} < f_{clk(sys)} \times 1.7$ (Hz)

Digital HF PLL: The digital PLL will recover the channel bit clock. The capture range of the PLL itself is very limited. To overcome this difficulty, two capture aids are present. When using automatic locking, the PLL will switch states based on the difference between expected distance and actual distance between synchronization.

Three different PLL operating modes exist:

- In-lock (normal operation): the PLL frequency matches the frequency of the channel bits with an accuracy error of less than 1 %
- Inner lock aid (capture aid 1): the PLL frequency matches the frequency of the channel bits with an accuracy error of between 1 % and 10 %
- Outer lock aid (capture aid 2): the PLL frequency is more than 10 % away from the channel bit frequency

The operation of PLL in-lock (normal on-track situation) is explained below followed by lock-detection and then the two capture aids.

PLL in-lock characteristics: The PLL behavior during in-lock can best be explained in the frequency domain. PLL operation is completely linear during in-lock situations. The open-loop response of the PLL (bode diagram) is given in Figure 17.

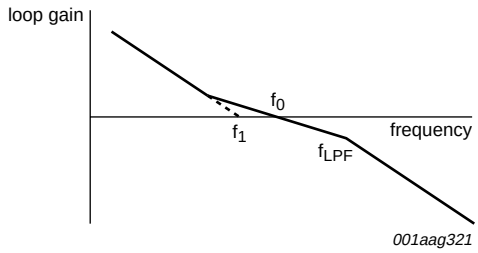


Fig 17. PLL bode diagram

f_1: Integrator cross-over controlled via K_I

f_0: PLL bandwidth controlled via K_P

f_2: LPF bandwidth controlled via K_F

The three frequencies are programmable using register PLLBandWidth. The higher bandwidths for use after a defect, are programmed in register PLLBandWidthHigh; see Section "Defect detector" on page 26.

When the PLL is in lock, the recovered PLL clock frequency equals the channel bit clock frequency.

Detection of PLL lock: The PLL locking state is determined by the distance between detected syncs. This means that the sync detection is actually controlling the automatic PLL locking.

The PLL switches from outer lock to inner lock when successive syncs are detected to be 588 ± 25 channel bits apart. Internally this is also called a 'winsync' (sync falls in a wider window). The number of missed winsyncs is kept in a 3-bit confidence counter, and the PLL will go out of outer lock when seven consecutive out-of-window syncs are found.

The PLL switches from inner lock to in-lock when successive syncs are detected 588 ± 1 channel bits apart. The number of consecutive missed syncs is kept in a bit counter, and saturates on either 16 or 61, depending on the value of bit LOCK[16] or [61] in register DemodControl. When the saturation level is reached, the PLL is set out-of-lock.

The PLL frequency (inner-) and phase (in-) lock status can be read out in register PLLLockStatus.

PLL outer-lock aid: The outer lock aid has no limitation on capture range, and will bring the PLL within the range of the inner lock aid. The PLL will first regulate its frequency based on detecting RL3s as the smallest possible RLs (fast-but-rough regulation), and next on detecting RL11s as the largest possible RLs (slow but more accurate).

PLL inner-lock aid: The inner-lock aid has a capture range of ± 4 %, and will bring the PLL frequency to the phase-lock point. It will regulate the PLL frequency such that 588-bits are detected between two EFM-syncs.

Influencing PLL behavior: Programmability and observer ability are built into the PLL mainly for debugging purposes, and also to make difficult applications possible. The PLL operation can be influenced in two ways:

- optional manual selection of the PLL state (in-lock, inner-lock, outer-lock, outer-lock with only RL3 regulation).
- optional pre-set of the PLL frequency to a certain value

Overriding the PLL state: PLL state can be:

- In-lock
- Inner lock
- Outer lock
- Outer lock with only RL3 regulation
- Hold

Normally, the PLL state is selected automatically by the lock detectors. However, the PLL lock state can be overruled via register PLLLockAidControl. When Lock mode is left at '0', the user can still choose the PLL state, but hardware will overwrite this if the hardware-selected PLL state is closer to locking the PLL.

Table 5. PLL lock states

Lock mode	PLLLockControl	State[1]
0	0 0000	automatic lock behavior
1	0 0001	force HF PLL into in-lock
1	0 0110	force HF PLL into inner-lock aid
1	0 0100	force HF PLL into outer-lock aid
1	0 1000	force HF PLL into hold mode
1	1 0100	force HF PLL into outer-lock aid with RL3 regulation only
x	others	reserved

[1] During PLL hold, the PLL frequency will not change and the frequency preset may be used.

It is possible to pre-set the PLL frequency to a certain value by writing the integrator value of the PLL to register PLLIntegrator. The relationship between the bit frequency, the integrator value, and the sysclk frequency $f_{clk(sys)}$ is given by:

$$f_{clk(bit)ch} = \frac{PLLFreq[7:0] + 4}{128} \times f_{clk(sys)}\ (Hz) \quad (2)$$

The real-time value of the PLL frequency can be read at the same address.

6.5.5.3 Limiting the PLL frequency range

The range over which the PLL can capture the input frequency can be limited. The minimum and maximum PLL frequency can be set by bits MININTFREQ, and MAXINTFREQ respectively in register PLLMinMaxBounds.

6.5.5.4 Run length 2 push-back detector

If this circuit is switched on, all run length one and two symbols (invalid run lengths) are pushed back to run length 3. For RL2s, the circuit will determine the transition that was most likely to be in error, and shift transition on that edge. This feature should always be turned on, but can be deselected via register RL2PushBack.

6.5.5.5 Available signals for monitoring

The operation of the bit detector can be monitored by the microcontroller via an external pin. Five signals are available for measurement:

PLL frequency signal: The microcontroller monitors this signal by reading register PLLIntegrator.

Asymmetry signal: This signal is in 2's complement form and can be read from register SlicerAssym.

Jitter signal: A jitter measurement is done internally. The zero-crossing jitter value is available in register PLLJitter.

Internal lock flags.

Jitter signal:

Jitter measurement is done in two steps:

1. The distance between the EFM zero transition and the bit clock zero transition is measured.
2. The calculated jitter for the zero transition is averaged using a 10-bit low-pass filter. The top 8-bits of the filter output can be read back from register PLLJitter. To obtain the jitter in % of the channel bit clock, Equation 3 applies:

$$jitter = \sqrt{\frac{jitter[7:0] - 2.83}{1024}} \times 100\ (\%) \quad (3)$$

Table 6. Jitter input calculation

Distance ($\times f_{bit}$)	Average distance (bit clocks)	Jitter filter input (5-bit decimal integer)
$< 2/16$	$1/16$	1
$2/16$ to $4/16$	$3/16$	9
$4/16$ to $6/16$	$5/16$	25
$> 6/16$	$7/16$	49

This jitter measurement is also available via the Meas1 telemetry signal on pin CL1. The full 10-bit output of the filter is available via this pin; see Section 6.5.5.8.

It is also possible to read out an average jitter value via register PLLAverageJitter. This value is an average of the normal jitter value over a period of 8000 bit clock periods. The formula to transform this into a percentage is shown in Equation 4:

$$average\ jitter = \sqrt{\frac{averagejitter[7:0] - 2.83}{1024}} \times 100\ (\%) \quad (4)$$

6.5.5.6 Use of jitter measurement

The jitter measurement is an absolute-reference jitter measurement. It gives the average square value of the bit detection jitter. Note that bit-to-clock jitter is measured. In this device, the bit-to-clock jitter is measured directly before the bit detection, and contains contributions due to various imperfections of the complete signal path:

- Disc

- Analog preamplifier
- ADC
- Limited bandwidths in this device
- Limited PLL performance
- Influenced by internal noise filter, asymmetry compensation and equalizer

The jitter measurement is absolute reference, because it relates directly to the EFM bit error rate if the disc noise is gaussian.

6.5.5.7 Internal lock flags

The fourth signal that can be monitored are three flags in the PLLLockStatus register: the internally generated inner lock signal FLock, the internally generated lock signal InLock and a LongSym(bol) flag when run length 14 is detected (run length too high).

In automatic mode, the FLock and InLock flags determine what type of PLL capture aid is used.

Table 7. Determining the current PLL capture mode

FLock flag	InLock flag	Capture mode
0	0	outer-lock aid
1	0	inner-lock aid
x	1	in-lock

6.5.5.8 Format of the measurements signal Meas1 on pin CL1

This signal is output via pin CL1 (pin 97) and comprises three measurement signals multiplexed together. The format is shown in Figure 18 and Table 8.

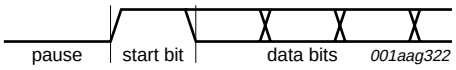


Fig 18. Format on measurement pin CL1

The data is sent in a serial format. It consists of a pause, followed by a start bit, followed by data bits.

Bit length: four system clock periods; frame length: 64 bits.

Table 8. Data format on measurement pin CL1

Bit	Value	Description
0	'1'	start bit[1]
1 to 10	jitter(9) to jitter(0)	first sample of jitter word[2]
11	'0'	
12	'1'	intermediate start bit
13 to 22	pllfreq(9) to pllfreq(0)	PLL frequency word
23	'0'	
24	'1'	intermediate start bit
25 to 32	asym(7) to asym(0)	slicer level
33, 34, 35	'0'	'000'

Table 8. Data format on measurement pin CL1 *…continued*

Bit	Value	Description
36	'1'	intermediate start bit
37 to 46	jitter(9) to jitter(0)	second sample of jitter word[2]
47 to 63	'0'	pause

[1] The start bit is always preceded by 17 pause bits. The intermediate start bits at bit locations 12, 24 and 36 guarantee that no other '1' value is preceded by 17 '0' bits. This allows a simple start bit detection circuit.

[2] The jitter word is sampled twice in every frame.
The percentage jitter is calculated using Equation 5:

$$jitter = \sqrt{\frac{jitter[9:0] - 12.81}{4096}} \times 100\ (\%) \quad (5)$$

6.5.5.9 Demodulator

The demodulator block performs the following functions:

- EFM demodulation using a logic array
- Sync detection and synchronization
- Sync protection

6.5.5.10 EFM demodulation

Each EFM word of 14 channel bits (which are separated from each other by three merging bits) is demodulated into one data byte making use of the standard logic array demodulation as described in the *CD Red Book (IEC 60908).*

6.5.5.11 Sync detection and synchronization

The EFM sync pattern is a unique pattern which is not used anywhere else in the EFM data stream. It consists of 24 bits: RL11 + RL11 + RL2. An internal sync pulse is generated when two successive RL11s are detected. A sub-sync pulse is produced when the beginning of a new subcode frame is seen. This is done by analyzing the subcode information: when two successive subcodes are subcode sync-code S0 and S1, sub-sync will be activated.

6.5.5.12 Sync protection

The sub-sync pulse is protected by an interpolation counter, this counter uses the fact that a subcode frame is always 98 subcode symbols long.

The sync signal itself is also interpolated. If after 33 data bytes (one EFM frame), no new sync is detected, it is assumed that the bit detector has failed to correctly produce it, and the sync signal is given anyway, this is generally called an 'interpolated sync.' Furthermore, if a new sync is detected in the data shortly after a previous sync signal, interpolated or real, no new sync signal will be produced, because this means the frame has 'slipped'. After enough data byte periods, the sync signals are allowed to pass again.

Although the possibility is small, 'false syncs' can be detected, such as corrupted EFM bits that accidentally form the combination RL11 + RL11. If two, or three, of such false re-syncs are detected at the correct distance from each other, this would cause a false sync of the demodulator. Such a re-sync could lead to a large number of samples being corrupted at the output of the CIRC decoder. The chance of false sync detection is greatest during defects (black and white dots).

To prevent such false demodulator re-syncs, two features are built in, which are both programmable via register DemodControl:

- RobustCntResync: This feature should always be turned on: when it is on, the demodulator will look for three consecutive syncs instead of two, with correct in-between distance before re-syncing. This should greatly improve the robustness against false syncs.
- SyncGating: when ‘1’, the sync detection is turned off during a defect, to avoid the detection of false syncs; when ‘0’, sync detection is left on permanently. Note that the defect detector needs to be set-up properly before this feature can be used. For this reason, this feature is turned off by default after reset.

6.5.6 CD decoding

6.5.6.1 General description of CD decoding

The decoder block performs all processing related to error correction and CIRC de-interleaving and makes use of an internal SRAM FIFO which provides the necessary data capacity. It also extracts the Q-channel subcode and the CD-text information from the data stream and delivers it to the application via a register interface.

6.5.6.2 Q-channel subcode interface

The channel decoder contains an internal buffer which stores the Q-channel bytes of a CD-subcode frame. This subcode can be retrieved by the microcontroller, by accessing the registers SubcodeQStatus, SubcodeQData and SubcodeQReadend.

To start retrieving the subcode, the microcontroller must first read the register SubcodeQStatus. This register contains various status bits that indicate the status of the Q-subcode that may be read. When, after reading the register SubcodeQStatus, the QREADY bit is found HIGH, the Q-subcode interface will be blocked (indicated by QBUSY going HIGH) so that no new subcode will overwrite the current one. QCRCOK indicates if the current subcode frame was indicated correctly, or not, by a hardware CRC check.

After reading SubcodeQStatus with QREADY = '1', the microcontroller may retrieve as many subcode bytes as required (10 maximum) by issuing subsequent reads to register SubcodeQData.

The content of the Q-channel subcode in the main data area is described in Table 9. For a description of the content during the lead-in area, refer to the *CD Red Book (IEC 60908)*.

Table 9. Q-channel subcode frame content

Address/Byte	Name	Comments	
1	CONTROL/MODE		
2	TNO		
3	POINT		
4	REL MIN	Mod100	relative time
5	REL SEC	Mod 60	
6	REL FRAME	Mod 75	
7	ZERO	0 or incremented modulo 10	

Table 9. Q-channel subcode frame content *…continued*

Address/Byte	Name	Comments	
8	ABS MIN	Mod100	absolute time
9	ABS SEC	Mod 60	
10	ABS FRAME	Mod 75	

After finishing a subcode read, the microcontroller must release the interface to allow the decoder to capture new subcode information. This is done by issuing a read to register SubcodeQReadend.

The availability of a new subcode frame will also trigger an interrupt if InterruptEnable2 bit SUBCODEREADYENABLE is set.

6.5.6.3 CD-text interface

The channel decoder contains an internal buffer which stores CD-text information (format 4, available in the lead-in area). The buffer can hold one CD-text pack for readback, while at the same time, it receives the next pack.

The operation of the CD-text readback interface is controlled via register CDTEXTControl. Bit FREEZEEN determines whether or not the internal buffer is frozen during readback (such that the next pack cannot overwrite the current one before the microprocessor has finished reading). Bit CRCFAILEN determines whether or not packs with a failed CRC check are made available for readback.

This subcode can be retrieved by the microcontroller, by accessing the registers CDTEXTStatus, CDTEXTData and CDTEXTReadEnd.

To start retrieving the CD-text pack, the microcontroller must first read the register CDTEXTStatus. This register contains various status bits that indicate the status of the CD-text pack that may be read. When, after reading the register CDTEXTStatus, the TEXTREADY bit is found HIGH, the CDTEXT interface will be blocked (indicated by TEXTBUSY going HIGH) so that no new subcode will overwrite the current one; at least if CDTEXTControl bit FREEZEEN is turned on. TextCRCOK indicates if the current CD-text pack was indicated correctly, or not, by a hardware CRC check.

After reading CDTEXTStatus with TEXTREADY = '1', the microcontroller may retrieve as many CD-text bytes as required (16 maximum) by issuing subsequent reads to register CDTEXTData.

After finishing CD-text read, the microcontroller must release the interface to allow the decoder to capture new CD-text information. This is done by issuing a read to register CDTEXTReadEnd.

Remark: If CDTEXTControl bit FREEZEEN is disabled, the interface is not held during readback, which means it is possible that the current CD-text pack is overwritten by the next one before all bytes of the current pack are read out. Such an event will be indicated by setting CDTEXTReadEnd bit BUFFEROVERFLOW HIGH, so that it can be noticed by software at the end of the pack read.

The availability of a new CD-text pack will also trigger an interrupt if InterruptEnable1 bit CDTEXTREADYENABLE) is set.

6.5.7 Main data decoding

6.5.7.1 Data processing

The CD main data is de-interleaved and error-corrected according to the *CD Red Book (IEC 60908)* CIRC decoding standards, and uses an internal SRAM as buffer and FIFO. The C1 correction will correct up to two errors per EFM frame, and will flag all uncorrectable frames as an erasure. The C2 error correction will correct up to two errors or four erasures, and will also flag all uncorrectable frames as an erasure.

The decoding operation is controlled by register DecoMode. There are basically two decode operating modes:

- Flush mode: the de-interleaver tables are emptied, and all internal pointers are reset. No data is written into the buffer, no corrections are done, and no data is output
- Play mode: de-interleaver tables are filled, C1 or C2 corrections are done, and data is output, when available

During Flush mode, no data is output from the device. During Play mode, data is output via the I^2S interface as soon as it is available in the internal FIFO.

Figure 19 shows the operation of the FIFO and corrections during CD playback.

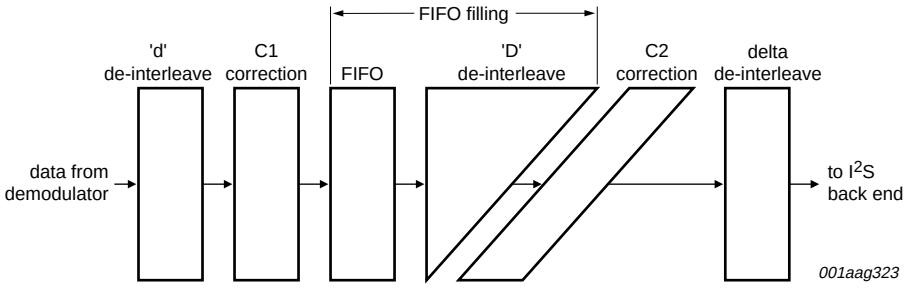


Fig 19. Data processing during CD mode

De-interleaving of the data is done as required by the *CD Red Book (IEC 60908)* specification. De-interleaving is performed by the SRAM FIFO address calculation functions in the memory processor. Two corrections are done: C1 followed by C2.

6.5.7.2 Data Latency + FIFO operation

System data latency is a function of the minimum amount of data required in the FIFO to perform the de-interleaving operation. The latency is quoted in the number of C1 frames (24 bytes of user data). The latency of the CIRC decoder is 118 frames.

The FIFO filling is defined as this ‘data latency’ + the number of extra frames stored in the FIFO. The filling of the FIFO must be maintained within certain limits. 118 frames is the minimum required for de-interleaving, 128 is the physical maximum limit determined by the used SRAM size. This results in a usable FIFO size of 11 frames. The FIFO filling can be read back via register FIFOFill.

The FIFO filling must have a correct value. This can be achieved in two ways:

- Master (Flow Control) mode: selected when using a gated bit clock (bclk) at the I^2S interface, see Section 6.5.9.9 on page 43 for more information. When a frame is available in the FIFO, it is output via the I^2S interface. When FIFO underflow is imminent, the decoder gates the output interface by disabling bclk.
- Slave (audio) mode: the bclk is continuously clocked in this mode. The application is responsible for matching the input rate (EFM bit rate from the disc) to the selected output rate (I^2S bclk speed), keeping FIFO filling between 118 and 128 frames. This is done by regulating the disc speed. See Section 6.5.10 for more details.

The FIFO only stores data, not subcode. This means that the data will be delayed as it comes from the demodulator, but the subcode is sent directly over the I^2S interface. The difference in delay between subcode and data is always fixed. It is absolutely fixed in Master mode, but can have small local variations in Slave mode.

6.5.7.3 Safe and unsafe correction modes

The CD CIRC decoding standard uses a Reed-Solomon (RS) error correction scheme. Reed-Solomon error correction has a very small chance of miscorrection, where a corrupted code word is modified into a valid but wrong code word. This results in the code word, after correction, being a valid existing RS code word but not the word that used to be present at this location before corruption. The chance of miscorrections increases exponentially for every extra byte that needs to be corrected in a code word, and is greatest when performing the maximum number of corrections possible with a certain RS correction scheme.

Miscorrections should be avoided, since they result in corrupted data being sent to the back end, without their corresponding invalid flag being set. This is a problem for CD audio, as unflagged wrong data will not get interpolated, which can result in audible clicks.

Both C1 and C2 correction logic can be programmed to operate in a ‘unsafe’ or ‘safe’ mode via register ErcoControl. In unsafe mode, the maximum number of corrections will always be done (if required). In Safe mode, corrections will not be done when they are considered too ‘unsafe’, which means there is a realistic chance that they could lead to a miscorrection.

For C1, unsafe mode allows two bytes per code word to be corrected, Safe mode allows only one byte per code word. For C2, both modes will allow up to four erasures per code word to be corrected. When there are more than four erasures, and therefore ERCO switches back to error correction, unsafe mode allows two bytes to be corrected, Safe mode allows only one.

Remark: From experiments and theory it is advised to use C1 unsafe and C2 safe for CD audio as a good trade-off between safety and maximum error correction capability. For CD-ROM, use C1 unsafe and C2 unsafe if there is at least a C3 error correction and if the flywheels in the CD-ROM block decoder are robust against possible invalid, but not flagged, headers.

6.5.8 Error corrector statistics

6.5.8.1 CFLG

The error corrector outputs status information on pin CFLG. The format of this information is serial and similar to the Meas1 signal on pin CL1.

The serial format consists of a pause bit followed by a start bit, followed by data bits. The format of the data is explained in Table 10.

Bit length: seven sysclk periods.

Frame length: 11 bits.

Table 10. Format description of CFLG serial bus

Bit	Value	Meaning	Note
0	1	start bit	[1]
1 to 3	Cormode[2:0]	type of correction	[2]
4	FlagFail	failure flag set because correction too unsafe	[3]
5	CorFail	failure flag set because correction impossible	[3]
9, 6 to 8	ErrorCount[3:0]	number of errors corrected	[4]
10	0	pause bit	[1]

[1] The CFLG repetition rate is not fixed and depends on disc speed and output interface speed. There is always at least one pause bit.

[2] Cormode definition: '000': C1 correction, '011': C2 correction, '100': corrector not active, Others: not used.

[3] FlagFail and CorFail indicate failure status on previous code word.

[4] ErrorCount indicates the number of errors found by the error corrector.

6.5.8.2 BLER counters

There are two BLER counters which count the number of frames C1 and C2, with at least one error. C2 erasures coming from C1 are also counted; it is irrelevant if the frame was correctable or not.

These registers are reset on read, and the user is responsible for reading them at regular intervals. The BLER counters can be read by registers C1Bler and C2Bler.

6.5.9 Audio back end and data output interfaces

The channel decoder back end is shown in Figure 20.

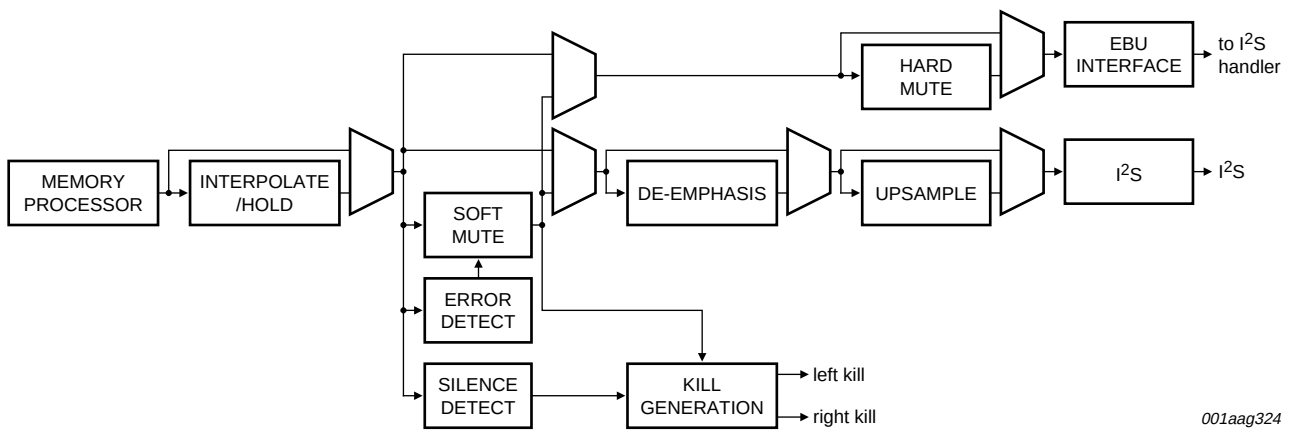


Fig 20. Backend audio functions

Decoded and error-corrected CD data streams into the back end from the memory processor to the output interfaces; some audio filtering can be done in-between, when playing CD-DA for example.

6.5.9.1 Audio processing

The following audio features are present in the back end:

- Interpolate/hold for I^2S
- Soft mute for I^2S
- De-emphasis filter for I^2S
- Upsample filter for I^2S
- Error detection
- Silence detection
- Kill generation

Some status bits concerning these audio features can be read back via register MuteKillStatus.

6.5.9.2 Interpolate-and-hold

On CD audio discs with many large defects, where C1 or C2 correction is unable to correct all the errors, the audio data can be interpolated or held, to avoid audible clicks or plops when playing back the disc. This feature is enabled by setting FilterConfig bit INTERPOLATEEN.

The interpolate-and-hold principle is shown in Figure 21.

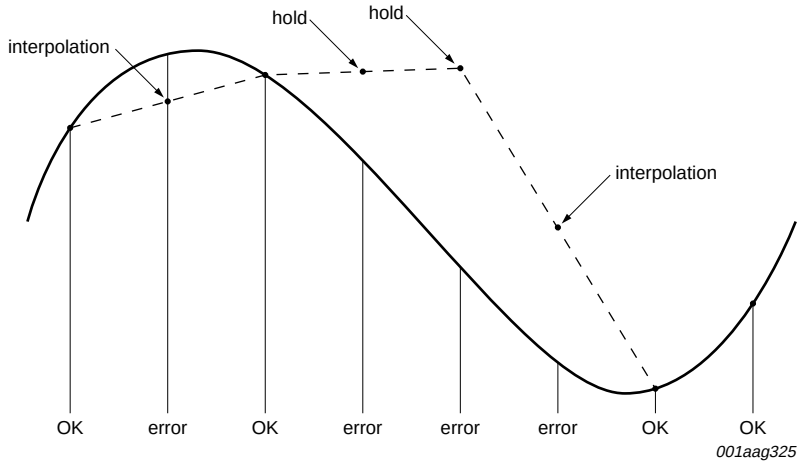


Fig 21. CD audio error concealment

Audio samples flagged as uncorrectable, neighbored by two good samples, or a held and a good sample, will be interpolated. Audio samples flagged as uncorrectable, which are not followed by a good sample, will hold the previous (correct or held) sample value.

This feature is enabled or disabled for I^2S and EBU together.

6.5.9.3 Soft mute and error detection

The audio data going to the I^2S and/or EBU interface can be processed by a soft mute block. This block can ramp the audio volume down from 0 dB to –90 dB, making use of 64 stages of about –1.5 dB each. The current stage can be monitored and changed by software read or write register MuteVolume. This allows the implementation of a software mute scheme. If the hardware mute logic is triggered by the error detection block (explained in the next paragraph), it ramps the volume down from maximum until fully muted in 3 / N ms, where N is the X-rate of the disc. The mute logic can be enabled separately for the I^2S and EBU outputs, by setting the corresponding bits in register MuteConfig.

The back end also contains an error detection block, that scans the data for a programmable number (via register MuteOnDefectDelay) of consecutive corrupted stereo samples. If such a pattern is found, and MuteConfig bit MUTEERREN is turned on, the soft mute will be triggered to start its volume ramp down. This detection will also trigger an InterruptStatus1 bit AUDIOERRORDETECTED interrupt.

6.5.9.4 Hard mute on EBU

The EBU can be hard muted (EBU main data and flags set to 0, status and user channel still valid) by setting EBUConfig bit EBUHARDMUTE.

6.5.9.5 Silence detection and kill generation

The silence detector looks for 250 ms of digital silence (2's complement data = all '1s' or all '0s') on either one or both channels and can trigger the kill logic when it is found. This feature is enabled via KillConfig bit KILLSILENCEEN.

The kill logic generates left and right kill signal outputs from the channel decoder that can be used to gate the left and right channels of an audio DAC. The kill signals can be triggered on both channels together by the detection of stereo silence, or on each channel separately by the detection of mono silence. The operation that is active depends on the setting of register KillConfig. It is also possible to set the left and right kill signals by software writing directly to bits KILLLEFT and KILLRIGHT in this register.

Another condition that sets both left and right kill signals is the soft mute block reaching 'fully muted' (volume-stage 0).

6.5.9.6 De-emphasis filter

This feature only affects the I^2S, not the EBU output. The de-emphasis filter can be used to remove pre-emphasis from tracks which have been recorded using the standard emphasis as described in the *CD Red Book (IEC 60908)*. The de-emphasis filter has the inverse response of the emphasis characteristics as described in the standard.

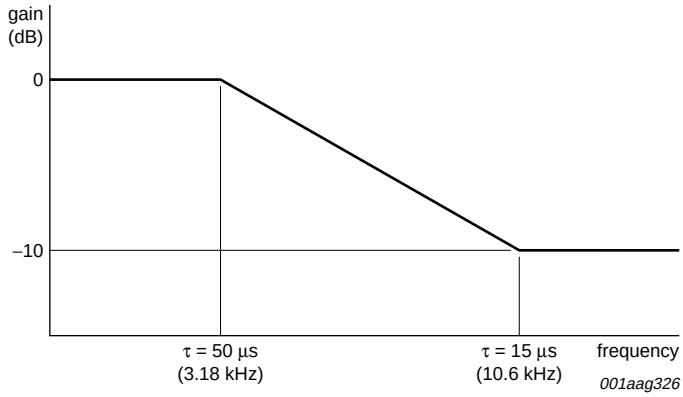


Fig 22. De-emphasis characteristics

The de-emphasis filter is controlled via FilterConfig bit DEEMPHCONTROL. The filter can be enabled or disabled under software control, or automatically by hardware. In the latter case, the filter is turned on when a 'pre-emphasis' bit is detected in the control byte of the Q-channel subcode, and turned off when this bit is missing.

There are two detection modes possible:

- According to the *CD Red Book (IEC 60908)*: the 'pre-emphasis' bit is only checked during the lead-in area (allowed to change), and during pauses between tracks
- According to the *CD Orange Book*: pre-emphasis is checked on every subcode frame

6.5.9.7 Upsample filter (four times)

This feature only affects the I^2S, not the EBU output. When it is enabled, the audio data is upsampled by a factor of four. The upsampling provides the frequency response described in Table 11.

Table 11. Upsample filter frequency response

Passband	Stop-band	Attenuation
0 Hz to 9 kHz	-	≤ 0.001 dB
9 kHz to 20 kHz	-	≤ 0.03 dB
-	24 kHz	≥ 25 dB
-	24 kHz to 27 kHz	≥ 38 dB
-	27 kHz to 35 kHz	≥ 40 dB
-	35 kHz to 64 kHz	≥ 50 dB
-	64 kHz to 68 kHz	≥ 31 dB
-	68 kHz	≥ 35 dB
-	69 kHz to 88 kHz	≥ 40 dB

When upsampling is enabled, the audio data output rate on the I^2S interface is four times higher than without upsampling. Therefore, the I^2S wclk frequency has to be four times higher. This means the I^2S bclk speed needs to be programmed to be four times higher than normally required for the X-rate when upsampling would be disabled.

Another result of the upsampling is that every sample will have 18-bit precision after the upsample filter instead of 16-bit. To make use of this extra bit precision, the user should select I²S-24 of -32 format. When using I²S-16 format, the two lowest bits will not be output.

6.5.9.8 Data output interfaces

There are three interfaces via which data can be output from the channel decoder block.

- I²S: main data
- subcode (V4) interface: subcode
- EBU or S/PDIF: main data + subcode

All interfaces can be used at the same time if needed, although there are a few restrictions on the EBU, see Section 6.5.9.10 on page 44.

6.5.9.9 I²S interface

The I²S is a 6-wire interface (four main + two subcode). It supports 16-bit, 24-bit and 32-bit I²S and EIAJ (Sony) modes. Timing is shown in Figure 23. The required format can be selected in register I2SFormat.

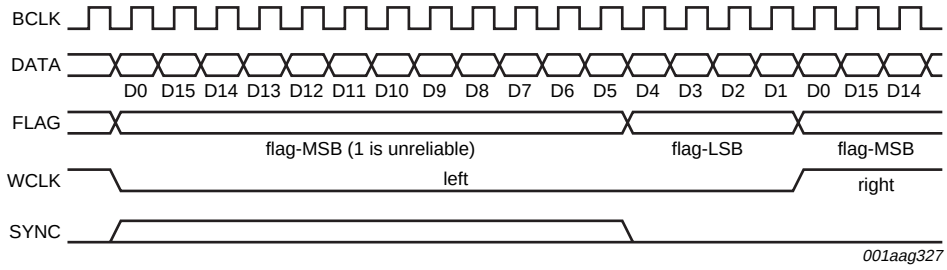


Fig 23. I²S format 1 (16 clocks per word)

In compliance with the I²S specification, I²S wclk, I²S data, I²S flag and I²S sync are all clocked on the falling edge of I²S bclk.

- I²S bclk: all other I²S signals are clocked on I²S bclk
- I²S wclk: indicates the start of a new 16- or 18-bit word on the data line, + distinction between left and right sample
- I²S data: 16-bit or 18-bit data words are output via this line, 1-bit or bclk period
- I²S flag: contains the byte reliability flag; bytes that are indicated as erasures (possible errors) after C1 and C2 correction, are flagged
- I²S sync: Indicates that the serial subcode line (V4) contains the MSB of a subcode word; it will be asserted for half a wclk-period after every six wclk-periods. If a subcode sync is transferred on the subcode line, this signal will be asserted for a full wclk period.

The I²S interface can either work in Master or Slave mode. In Master mode, the I²S bclk can be gated off by the channel decoder. In Slave mode, the I²S bclk is continuously running. To prevent the internal FIFO from overflow, the filling of the buffer must be regulated (see Section 6.5.7.2 on page 37).

I²S bclk and I²S wclk can either be input (generated outside the channel decoder) or output (generated internally in the clock control block). Selection can be done via WclkSel and BclkSel in register I2SConfig.

The I²S output rate is determined by the speed of the I²S bclk clock, which is configured via register BitClockConfig. One can configure the I²S interface to run at 1× CD speed or 2× CD speed.

For a gated bit clock, when BitClockConfig bit BCLKGEN is HIGH, the speed must be configured such that the maximum rate available on the bus is 20 % higher than the average data throughput rate i.e. the bus should have at least 20 % idle time between two bursts of data.

Default after reset, the I²S pins on the IC will be put into 3-state. They can be activated via register I2SConfig. This register also contains the possibility to 'kill' the I²S interface, such that all data lines output a constant '0'.

6.5.9.10 Subcode (V4) interface

Subcode data is output via the I²S subO (V4) port. This data can be sampled using the I²S sync signal (see Section 6.5.9.9 on page 43). The sync indicates that the serial subcode line (V4) contains the MSB of a subcode word; it will be asserted every six wclk-periods for half a wclk-period. If a subcode sync is transferred on the subcode line, this signal will be asserted for a full wclk period.

During normal operation (upsampling disabled), the subcode output via I²S subO will have the format as shown in Figure 24.

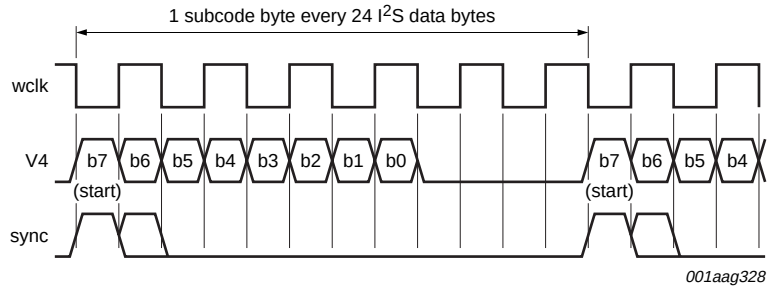


Fig 24. Subcode output (upsampling disabled)

When upsampling is enabled, the I²S interface runs at four times the non-upsampled rate. The subcode bit period however will stay at the bit period of the non-upsampled rate as shown in Figure 25. This means that the I²S subO and I²S sync signal will appear to be four times slower relative to I²S wclk. In this case the receiver must use I²S wclk divided by 4 to sample the subcode.

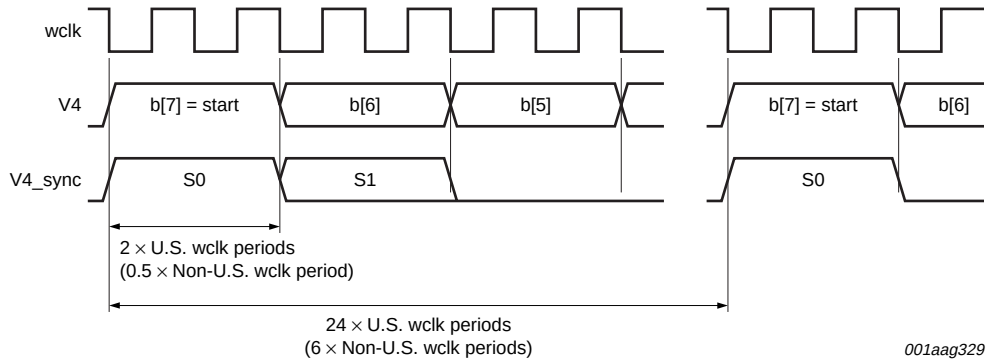


Fig 25. Subcode output (upsampling enabled)

When slave mode is used (no I^2S bclk gating), it is also possible to use the I^2S subO (V4) output port as a true single-line interface. In this case the receiver needs to sample the data on the line at a frequency I^2S wclk × 2 (Hz) (since subcode is output at a rate of 1-bit or half I^2S wclk). Two characteristics of the interface can be used in this case to synchronize the bit and byte detection in the stream in absence of an I^2S sync signal.

- The first bit (P-bit) of a subcode byte is used as a start bit and therefore always 1, (no real P-channel information available on the interface); between two subcode bytes there are four zero bits. This can be used to identify the start of the subcode bytes within the stream.
- The subcode syncs S0 and S1 are presented as all zeroes on the interface (even P-channel), such that the last subcode byte of a subcode frame, and the first byte of the next frame are separated by 28 zero bits. This can be used to identify the start of the subcode frames within the stream.

6.5.10 Motor

A block diagram of the motor interface is given in Figure 26.

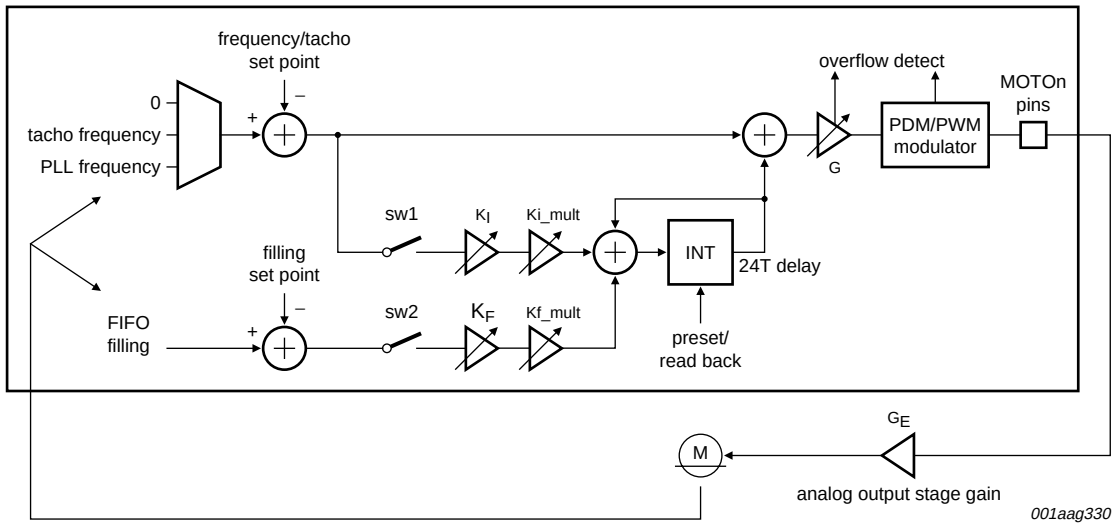


Fig 26. Motor servo block diagram

The motor servo consists of a PI filter and a PWM or PDM modulator. When put in a closed loop, the motor controller can control both speed/frequency and position error (FIFO fill). It can be operated as a P, I or PI controller, by switching on and off the appropriate switches (sw1, sw2).

The frequency and position error integrator gain, K_I and K_F, and gain G are programmable. Frequency and filling set points are also programmable.

The frequency input source can be selected between PLL frequency and zero. The position input source is always FIFO filling.

When operated in a stable operation point in closed loop, the motor controller regulates the frequency input source and the FIFO filling to their respective set points. This is implemented by speeding up or slowing down the motor by changing the DC content in the PDM/PWM output motor signals.

All parameters can be configured by programming the motor registers.

6.5.10.1 Frequency setpoint

When operating the motor in CLV mode, based on EFM, for a certain overspeed, the motor frequency set point to be programmed is given by:

$$MotorFreqSet[7:0] = 256 \times \left(1 - \frac{N \times 4.3218 \times 10^6}{2.667 \times f_{clk(sys)}}\right) \quad (6)$$

where $f_{clk(sys)}$ is the system clock frequency and N is the overspeed factor.

The set point can be programmed via register MotorFreqSet. The selection of the motor frequency input is programmed via MotorGainSet2 bit MOTORFREQSOURCE.

6.5.10.2 Position error

The position error will be used to fine tune the motor speed during Slave mode where the incoming EFM bit rate is locked on the programmed fixed I^2S bclk output speed. The set point must be chosen between 118 and 128, since this is the usable FIFO size in the decoder. For more information, see Section 6.5.7.2 on page 37.

The set point can be programmed via register MotorFifoSet.

6.5.10.3 Motor control loop gains (K_P, K_F and K_I)

The motor control loop gains are all programmable, through registers MotorGainSet1 and MotorGainSet2. To be able to set integrator bandwidth low enough at high system clock speeds, an extra divider for the factors K_I and K_F is added. These factors can be written through the register MotorMultiplier.

The resulting $K_{I(tot)}$ is then the K_I multiplied by Ki_mult. The resulting $K_{F(tot)}$ is then the K_F multiplied by Kf_mult. The integrator bandwidth must be scaled with the same factor Ki_mult.

Notes:

- Kf_mult operates by sampling the input. For example, for Kf_mult = 1, every sample of the input is passed through to the integrator circuit, for a Kf_mult of 0.5, every second sample is passed through, for a Kf_mult of 0.25, every fourth sample is passed through, and so on.
- For a DC input signal, $K_F \times$ Kf_mult should always give the same result. If however, the input is varying quickly, the $K_F \times$ Kf_mult combinations with the same product will not always give the same result, especially for low values of Kf_mult, where the sampling in the extreme becomes 1 out of every 128 samples. The input samples to the block that performs the Kf_mult multiplication occur at a rate of one sample every 24 system clock periods. Sub-sampling might affect the resulting gain.

6.5.10.4 Operation modes

The motor controller mode is programmed in register MotorControl. It can operate open loop, just sending a fixed power to the motor, for start-up and stopping, closed loop, or shut-down. This register also selects between PDM and PWM formats.

Motor start and stop modes will put a fixed duty cycle PWM or fixed density PDM signal on the motor outputs. During start or stop, motor speed can be monitored by reading MotorIntLSB and MotorIntMSB.

MotorOv: When not setting the appropriate gains in the loop, an overflow might occur inside the PWM/PDM modulator block, or in the programmable gain stage. This is signalled by the MotorOv interrupt, which can be read back on InterruptStatus2. The interrupt clears when the overflow clears.

MotorOv can also automatically open sw1 or sw2. This is enabled by writing a ‘1’ to bit OVFSW in register MotorControl.

6.5.10.5 Writing, reading motor integrator value

It is possible to obtain the integrator value by reading the registers MotorIntLSB and MotorIntMSB. The integrator can be written at the same location. By opening all switches, the user can bypass the whole control and filter part, and just use the block as a DAC towards the motor drivers. The control part can then be done by software.

6.5.10.6 Some notes on application motor servo

The motor servo can be used to control the motor during CLV playback and also during CAV or pseudo-CLV lock-to-disc or jump mode.

- In CLV mode, both sw1 and sw2 must be closed
- In CAV/pseudo-CLV mode, sw2 must be open and sw1 can be open
- The motor servo will revolve the disc at the speed corresponding to the frequency set point. In CLV mode with lock to EFM, the frequency set point must be selected equal to the desired readout frequency of the HF PLL
- Accelerating the disc must be done in one of the start modes
- Braking the disc must be done in one of the stop modes

6.5.10.7 Tacho

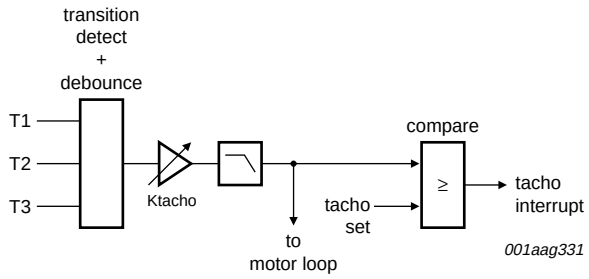


Fig 27. Tacho block diagram

The tacho circuit accepts the tacho input on the tacho inputs T1, T2 and T3, coming from the hall sensors on the spindle motor. The Tacho block measures the frequency of the input pulses, and filters the measurements using a second-order low-pass filter to remove noise on the tacho signal.

The measured tacho frequency can be read by the microprocessor, and can be used as a frequency input to the motor control, and can be used to generate the tacho trip frequency interrupt.

The relationship between the actual motor speed (in Hz) and the TachoFrequency[7:0] value read back in register TACHO4, bit TACHOFREQUENCY is given by:

$$TachoFrequency[7:0] = \frac{MotorFreq \times Ktacho[7:0] \times 2h \times p}{f_s} \quad (7)$$

Where:

- TachoFrequency[7:0] is the value read from register TACHO4; when doing CAV mode motor control, it is compared to the motor frequency set point programmed in register MotorFreqSet - an unsigned value
- MotorFreq is the angular velocity in rounds per second (Hz)
- Ktacho[7:0] is the gain value written to register TACHO1
- 2h: h is the number of hall sensors: 1 or 3, depending on motor and setting of bit ONEPINMODE in register TACHO3
- p is the number of motor pole-pairs, usually 1
- f_s is the sampling frequency of the filter; this can be configured via bit FSAMSEL in register TACHO4

Tacho gain Ktacho: The tacho gain, Ktacho[7:0] can be chosen so that the value read from bit TACHOFREQUENCY in register TACHO2 is the motor frequency in Hz. However, it is advisable to select Ktacho[7:0] such that a minimum amount of ripple is seen on the measured tacho frequency. This must be tuned experimentally.

Tests have shown that with good selection of Ktacho[7:0] and TACHOSAMPLERATE, a minimum amount of variation on the measured frequency can be obtained.

Tacho trip frequency: It is possible for the software to be notified with an interrupt when reaching a specific speed during spin-up or spin-down. This is done by programming the desired frequency trip point in register TACHO2. When the tacho frequency goes above or below this trip point, an interrupt gets generated (bit 3 of register InterruptStatus2). Bit 1 (TACHOINTERRUPTSELECT) of register TACHO3 can be set to enable an interrupt to be generated when the frequency goes above or below the trip point.

6.6 Parallel Digital Servo IC (PDSIC)

The digital servo block design on the SAF784x has evolved from the design used on the SAA7824 IC, and is referred to as the PDSIC. 'Parallel', refers to the microprocessor interface of the servo block which is now a high speed parallel interface. Previously, it was a serial interface used on the SAA7824, 3 or 4-wire, I^2C-bus. The PDSIC features are listed below:

- Programmable ADC for CD-RW playback compatibility
- Diode signal processing
- Signal conditioning
- Focus and radial control system
- Access control
- Sledge control
- Shock detector
- Defect detector
- Off-track counting and detection
- Automatic closed-loop gain control available for focus and radial loops
- Hi-level features
- Flexible servo

6.6.1 PDSIC registers and servo RAM control

The servo block is controlled by two parts of the design: the servo control registers which are used to control the writing of commands and parameters to the servo, and the servo RAM. The servo RAM has two roles: storage of the servo parameters, and capture of commands and parameters during the command process.

All of the servo write commands consist of a command byte followed by a number of parameter bytes (between one and seven), all of which have to be loaded into the PDSIC using a serial communication interface.

The command byte is the first to be loaded and can be considered as two nibbles. The upper (most significant) nibble represents the command itself whilst the lower (least significant) nibble tells the PDSIC how many parameter bytes to expect. The command byte gets placed into memory location 0x31 (called oldcom).

Subsequently, parameter bytes get loaded sequentially and these get placed into a stack space that has been reserved within the memory (locations 0x30 down to 0x2B). With each parameter byte that is loaded, the value in oldcom is decremented so that the byte count decreases to zero, and the PDSIC knows it has a complete servo command (a

command byte and its full compliment of parameter bytes). At this point, the PDSIC acts upon the command and the appropriate function is carried out based upon the values in the stack space.

There are two special case servo commands: Write_parameter (opcode = 0xA2) and Write_decoder_reg (opcode = 0xD1).

Write_parameter allows the microcontroller to write directly to any memory location. It carries two parameter bytes: the memory address and the data that is to be written. When this command is executed, the command byte is loaded into oldcom and the first parameter byte (RAM_address) is loaded in the stack. The second parameter byte (data) is loaded directly to the location specified by the RAM_address.

Write_decoder_reg allows decoder registers to be written to when the I^2C-bus interface is being used. This command carries only one parameter byte, which is the decoder register/data pair (two nibbles). When this command is received by the PDSIC, the register/data pair is loaded into memory location 0x4D.

The servo read commands operate slightly differently because they carry no parameter bytes and the lower nibble of the command byte is always 0 to indicate this. When the PDSIC receives a read command, it will make certain information available (mostly from memory, although some status information is retrieved from the decoder) on the serial interface for collection by the microcontroller.

If a sequence of values is being read from the servo RAM (e.g. a series of values related to a PID loop), it is important to ensure that the values are consistent with each other by ensuring that the servo has not updated some of the values during the period that they are being read. To prevent this occurring, an interrupt signal is available from the servo to the ARM which asserts an IRQ when it is safe to read related values. The interrupt generator monitors these signals and raises an IRQ whenever the correct state is achieved. The interrupt is cleared by applying a pulse to the Inreq_Clr register bit. If the interrupt is not cleared, it will automatically be reset when the valid reading state is no longer true.

Figure 28 shows the operation of the IRQ signal. Int #1 shows the full duration of an interrupt that does not get cleared by the ARM. Int #2 and Int #3 are shown being cleared by pulses being written to the Inreq_Clr register. The time between interrupts is approximately 15 µs and the total interrupt cycle time is about 60 µs.

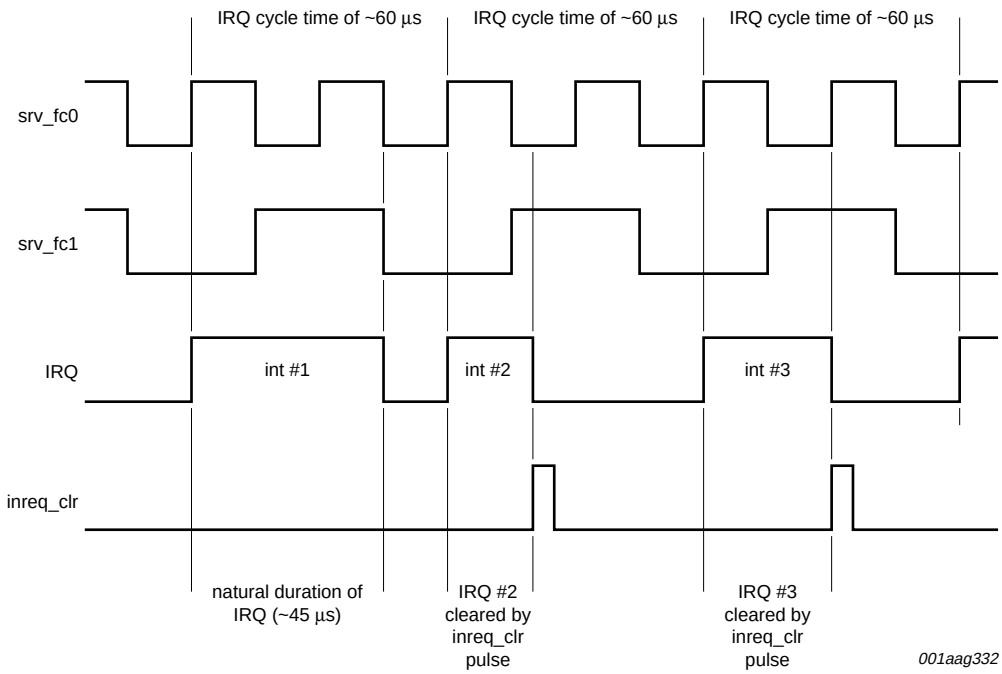


Fig 28. Function of servo IRQ signal with respect to srv_fc0, srv_fc1 and inreq_clr

The SAF784x contains additional circuits to implement a servo feature called Flexi servo.

The purpose of the flexible servo system is, in conjunction with the existing analog and digital LF path, to provide maximum flexibility in the use of the entire servo loop. This scheme extends from the point of diode PDM generation at the analog ADC outputs through to the servo actuator signals (RA, FO and SL) themselves.

From a system perspective, the simplest configuration provides an LF path that is equivalent to the hardware servo (PDSIC) of the other audio devices such as SAA7826, which uses the on-board hardware servo controller logic. However, an alternative setup will provide additional fine DC-offset compensation (in addition to the coarse compensation already found in the analog ADCs) and the potential for full software servo control via the ARM microprocessor. Full system configuration details are given in the block diagrams in the sections below.

6.6.2 Diode signal processing

The photo detector in conventional two-stage three-beam compact disc systems normally contains six discrete diodes. Four of these diodes (three for single foucault systems) carry the Central Aperture signal (CA) while the other two diodes (satellite diodes) carry the radial tracking information. The CA signals are summed into an HF signal for the decoder function and are also differenced (after A-D conversion) to produce the low frequency focus control signals.

The low frequency content of the six (five if single Foucault) photodiode inputs are converted to pulse-density modulated bit streams by a multiplexed 6-bit ADC followed by a digital PDM generation circuit. This supports a range of OPUs in Voltage mode mechanisms by having sixteen selectable gain ranges in two sets, one set for D1 to D4 and the other for R1 and R2.

6.6.3 Signal conditioning

The digital codes retrieved from the ADC and PDM generator are applied to logic circuitry to obtain the various control signals. The signals from the central aperture diodes are processed to obtain a normalized focus error signal.

$$FE_n = \frac{D1 - D2}{D1 + D2} - \frac{D3 - D4}{D3 + D4} \quad (8)$$

where the detector setup is assumed to be as shown in Figure 29.

For the single foucault focusing method, the signal conditioning can be switched under software control such that the signal processing is as follows:

$$FE_n = 2 \times \frac{D1 - D2}{D1 + D2} \quad (9)$$

The error signal, FE_n, is further processed by a Proportional-Integral and Differential (PID) filter section.

An internal flag is generated by means of the central aperture signal and an adjustable reference level. This signal is used to provide extra protection for the Track Loss (TL) generation, the focus start-up procedure and the dropout detection.

The radial or tracking error signal is generated by the satellite detector signals R1 and R2. The radial error signal can be formulated as follows:

$RE_s = (R1 - R2) \times re\_gain + (R1 + R2) \times re\_offset$

where the index ‘s’ indicates the automatic scaling operation which is performed on the radial error signal. This scaling is necessary to avoid non-optimum dynamic range usage in the digital representation and reduces the radial bandwidth spread. Furthermore, the radial error signal will be made free from offset during start-up of the disc.

The four signals from the central aperture detectors, together with the satellite detector signals generate a Track Position Indicator signal (TPI) which can be formulated as follows:

$TPI = sign\,[(D1 + D2 + D3 + D4) - (R1 + R2) \times sum\_gain]$

where the weighting factor sum_gain is generated internally by the SAF784x during initialization.

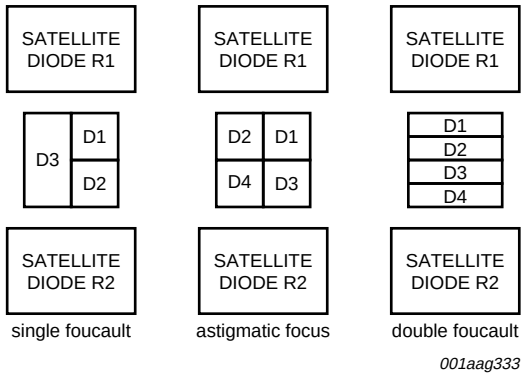


Fig 29. Detector arrangement

6.6.4 Focus servo system

6.6.4.1 Focus start-up

The start-up behavior of the focus controller is influenced by five initially loaded coefficients. The automatically-generated triangular voltage can be influenced by three parameters: height (ramp_height), DC offset (ramp_offset) of the triangle, and its steepness (ramp_incr).

For protection against detection of false focus points, two parameters are available which are an absolute level on the CA signal (CA_start) and a level on the FE_n signal (FE_start). Focus is achieved when this CA level is reached.

When focus is achieved and the level on the FE_n signal is reached, the focus PID is enabled to switch on when the next zero crossing is detected in the FE_n signal.

6.6.4.2 Focus position control loop

The focus control loop contains a digital PID controller which has five parameters available to the user. These coefficients influence integrating (foc_int), proportional (foc_lead_length, part of foc_parm3) and differentiating (foc_pole_lead, part of foc_parm1) action of the PID and a digital low-pass filter (foc_pole_noise, part of foc_parm2) following the PID. The fifth coefficient, foc_gain influences the loop gain. Figure 30 shows the transfer function of the controller, and the coefficients which determine its behavior.

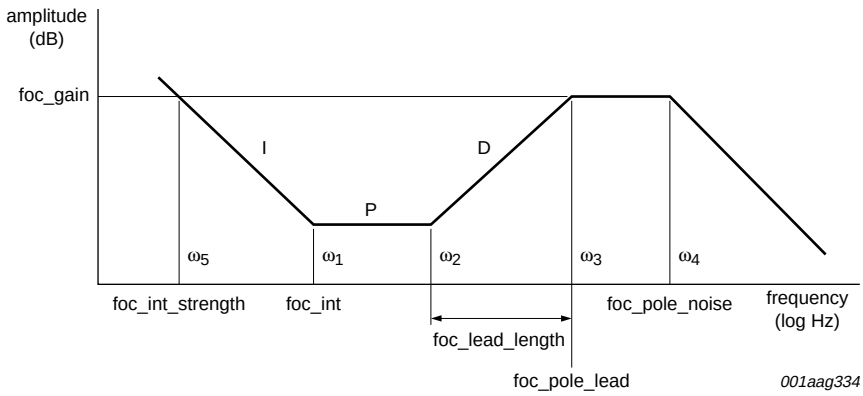


Fig 30. Bode diagram of focus PID system

A simplified block diagram of the focus PID system is given in Figure 31.

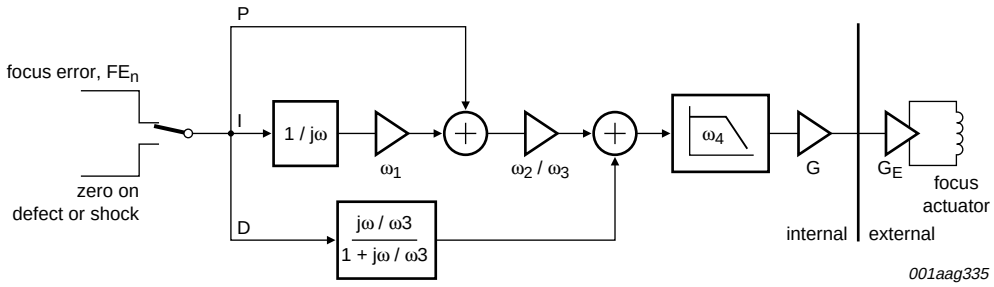


Fig 31. Block diagram of focus PID system

The actuator position can be held by using a zero error signal. This action is taken if a defect or shock is encountered. The PID is followed by a low-pass filter to reduce audible noise in the control loop.

The desired frequencies for the loop (ω_1 to ω_4) are used to calculate the coefficient values (full tables are given in the HSI). An explanation of the different parameters in these diagrams is given in Table 12.

Table 12. Focus PID parameters

Parameter	Controlled by	Comment
ω_1	-	focus integrator bandwidth
ω_2	-	start of focus lead
ω_3	foc_parm1	foc_pole_lead; end of focus lead (differentiating part)
ω_4	foc_parm2	foc_pole_noise; low-pass function following PID
ω_3 / ω_2	foc_parm3	foc_lead_length; lead length (proportional part)
$\omega_1 = (\omega_5 \times \omega_3 / \omega_2)$	foc_int_strength	integrator strength
G	foc_gain	focus loop gain
G_E	end stage gain	defined as peak-to-peak voltage swing over focus actuator

6.6.4.3 Dropout detection

This detector can be influenced by one parameter (CA_drop). Focus will be lost and the integrator of the PID will hold if the CA signal drops below this programmable absolute CA level. When focus is lost it is assumed, initially, to be caused by a black dot.

6.6.4.4 Focus loss detection and fast restart

Whenever focus is lost for longer than approximately 3 ms, it is assumed that the focus point is lost. A fast restart procedure is initiated which is capable of restarting the focus loop within 200 ms to 300 ms depending on the programmed coefficients of the microcontroller.

6.6.4.5 Focus loop gain switching

The gain of the focus control loop (foc_gain) can be multiplied by a factor of 2 or divided by a factor of 2 during normal operation. The integrator value of the PID is corrected accordingly. The differentiating (foc_pole_lead) action of the PID can be switched at the same time as the gain switching is performed.

6.6.4.6 Focus automatic gain control loop

The loop gain of the focus control loop can be corrected automatically to eliminate tolerances in the focus loop. This gain control injects a signal into the loop which is used to correct the loop gain. Since this decreases the optimum performance, the gain control should only be activated for a short time (for example, when starting a new disc).

6.6.5 Radial servo system

6.6.5.1 Radial PID - on-track mode

When the radial servo is in On-track mode (normal play mode), a PID controller is active for the fast actuator, while the sledge is steered using either a PI or pulsed-mode system. A simplified diagram of the radial PID system is given in Figure 32.

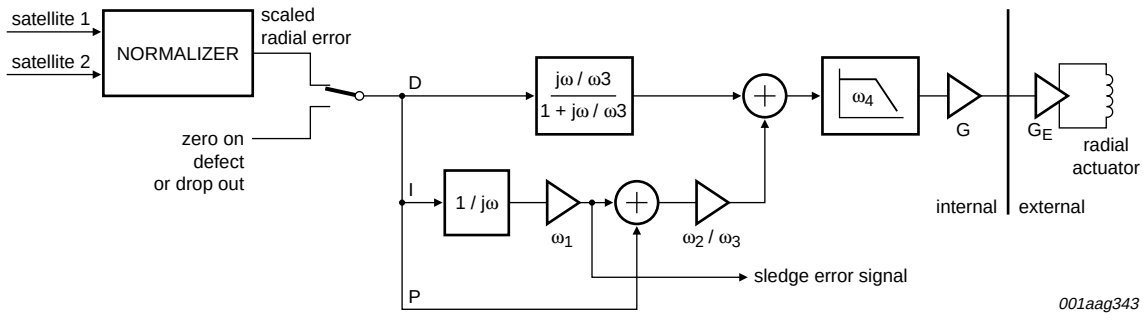


Fig 32. Block diagram of radial PID system

An explanation of the different radial PID parameters are given in Table 13. The system frequency response is given in Figure 33.

Table 13. Radial PID Parameters

Parameter	Controlled by	Comment
ω_1	-	radial integrator bandwidth
ω_2	-	start of radial lead
ω_3	rad_parm_play	end of radial lead (differentiating part)
ω_4	rad_pole_noise	low-pass function following PID
ω_3 / ω_2	rad_length_lead	lead length (proportional part)
$\omega_5 = (\omega_1 \times \omega_2 / \omega_3)$	rad_int_strength	integrator strength
G	rad_gain	radial loop gain
G_E	end stage gain	defined as peak-to-peak voltage-swing over radial actuator

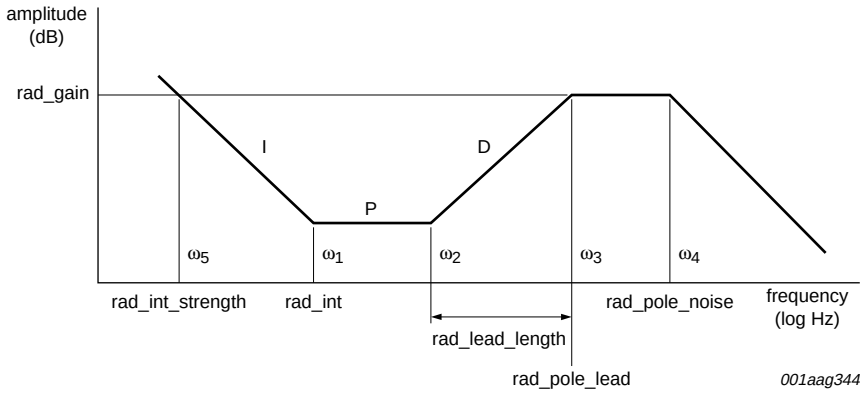


Fig 33. Bode diagram of radial PID system

6.6.5.2 Level initialization

During startup, an automatic adjustment procedure is activated to set the values of the radial error gain (re_gain), offset (re_offset) and satellite sum gain (sum_gain) for TPI level generation. The initialization procedure runs in a radial open loop situation and is ≤ 300 ms. This start-up time period may coincide with the last part of the motor start-up time period:

- Automatic gain adjustment: as a result of this initialization the amplitude of the RE signal is adjusted to within ± 10 % around the nominal RE amplitude
- Offset adjustment: the additional offset in RE due to the limited accuracy of the start-up procedure is less than ± 50 nm

TPI level generation: the accuracy of the initialization procedure is such that the duty factor range of TPI becomes 0.4 < duty factor < 0.6 (default duty factor = TPI HIGH / TPI period).

6.6.5.3 Dropout detection

This detector can be influenced by one parameter (CA_drop). Focus will be lost and the integrator of the PID will hold if the CA signal drops below this programmable absolute CA level. When focus is lost, it is assumed initially, to be caused by a black dot.

6.6.5.4 Focus loss detection and fast restart

Whenever focus is lost for longer than approximately 3 ms it is assumed that the focus point is lost. A fast restart procedure is initiated which is capable of restarting the focus loop within 200 ms to 300 ms depending on the programmed coefficients of the microcontroller.

6.6.5.5 Focus loop gain switching

The gain of the focus control loop (foc_gain) can be multiplied by a factor of 2 or divided by a factor of 2 during normal operation. The integrator value of the PID is corrected accordingly. The differentiating (foc_pole_lead) action of the PID can be switched at the same time as the gain switching is performed.

6.6.5.6 Focus automatic gain control loop

The loop gain of the focus control loop can be corrected automatically to eliminate tolerances in the focus loop. This gain control injects a signal into the loop which is used to correct the loop gain. Since this decreases the optimum performance, the gain control should only be activated for a short time (for example, when starting a new disc).

6.6.6 Radial servo system

6.6.6.1 Level initialization

During startup, an automatic adjustment procedure is activated to set the values of the radial error gain (re_gain), offset (re_offset) and satellite sum gain (sum_gain) for TPI level generation. The initialization procedure runs in a radial open-loop situation and is ≤ 300 ms. This start-up time period may coincide with the last part of the motor start-up time period:

- Automatic gain adjustment: as a result of this initialization the amplitude of the RE signal is adjusted to within ± 10 % around the nominal RE amplitude
- Offset adjustment: the additional offset in RE due to the limited accuracy of the start-up procedure is less than ± 50 nm
- TPI level generation: the accuracy of the initialization procedure is such that the duty factor range of TPI becomes 0.4 < duty factor < 0.6 (default duty factor = TPI HIGH / TPI period)

6.6.6.2 Sledge control

The microcontroller can move the sledge in both directions via the steer sledge command.

6.6.6.3 Tracking control

The actuator is controlled using a PID loop filter with user-defined coefficients and gain. For stable operation between the tracks, the S-curve is extended over 0.75 of the track. On request from the microcontroller, S-curve extension over 2.25 tracks is used, automatically changing to access control when exceeding those 2.25 tracks.

Both modes of S-curve extension make use of a track-count mechanism. In this mode, track counting results in an automatic ‘return-to-zero track’ to avoid major disturbances in the audio output and providing improved shock resistance. The sledge is continuously controlled, or provided with step pulses to reduce power consumption using the filtered value of the radial PID output. Alternatively, the microcontroller can read the average

voltage on the radial actuator and provide the sledge with step pulses to reduce power consumption. Filter coefficients of the continuous sledge control can be preset by the user.

6.6.6.4 Access

The access procedure is divided into two different modes depending on the requested jump size; see Table 14.

Table 14. Radial servo access procedure modes

Access type	Jump size	Access speed
Actuator jump	brake_distance[1]	decreasing velocity
Sledge jump	brake_distance - 32768	maximum power to sledge[1]

[1] Microcontroller presettable.

The access procedure makes use of a track-counting mechanism, a velocity signal based on a fixed number of tracks passed within a fixed time interval, a velocity set point calculated from the number of tracks remaining, and a user-programmable parameter indicating the maximum sledge performance.

If the number of tracks remaining is greater than the brake_distance then the sledge jump mode should be activated, or the actuator jump should be performed. The requested jump size together with the required sledge breaking distance at maximum access speed defines the brake_distance value.

During the actuator jump mode, velocity control with a PI controller is used for the actuator. The sledge is then continuously controlled using the filtered value of the radial PID output. All filter parameters (for actuator and sledge) are user programmable.

In the sledge jump mode, maximum power (user programmable) is applied to the sledge in the correct direction while the actuator becomes idle (the contents of the actuator integrator charge current reduces to zero just after the sledge jump mode is initiated). The actuator can be electronically damped during sledge jump. The gain of the damping loop is controlled via the hold_mult parameter.

The fast track jumping circuitry can be enabled or disabled via parameter xtra_preset.

6.6.6.5 Radial automatic gain control loop

The loop gain of the radial control loop can be corrected automatically to eliminate tolerances in the radial loop. This gain control injects a signal into the loop which is used to correct the loop gain. Since this decreases the optimum performance, the gain control should only be activated for a short time (for example, when starting a new disc).

This gain control differs from the level initialization. The level initialization should be performed first. The disadvantage of using the level initialization without the gain control is that only tolerances from the front end are reduced.

6.6.7 Off-track counting

The Track Position Indicator (TPI) signal is a flag which is used to indicate whether the radial spot is positioned on the track, with a margin of ± ¼ of the track-pitch. In combination with the Radial Polarity (RP) flag the relative spot position over the tracks can be determined.

These signals can have uncertainties caused by:

- Disc defects such as scratches and fingerprints
- The HF information on the disc; which is considered as noise by the detector signals

In order to determine the spot position with sufficient accuracy, extra conditions are necessary to generate a Track Loss (TL) signal and an off-track counter value. These extra conditions influence the maximum speed and this implies that, internally, one of the following three counting states is selected:

- Protected state: used in normal play situations; a good protection against false detection caused by disc defects is important in this state.
- Slow counting state: used in low-velocity track jump situations; in this state a fast response is important rather than the protection against disc defects (if the phase relationship between TL and RP of $\frac{1}{2}$ p radians is affected too much, the direction cannot then be determined accurately).
- Fast counting state: used in high-velocity track jump situations; highest obtainable velocity is the most important feature in this state.

6.6.8 Defect detection

A defect detection circuit is incorporated into the SAF784x. If a defect is detected, the radial and focus error signals may be zeroed, resulting in better playability. The defect detector can be switched off, applied only to focus control or applied to both focus and radial controls under software control (part of foc_parm1).

The defect detector has programmable set points selectable by the parameter defect_parm.

6.6.9 Off-track detection

During active radial tracking, off-track detection is realized by continuously monitoring the off-track counter value. The off-track flag becomes valid whenever the off-track counter value is not equal to zero. Depending on the type of extended S-curve, the off-track counter is reset after 0.75 track extend mode, or at the original track in the 2.25 track extend mode.

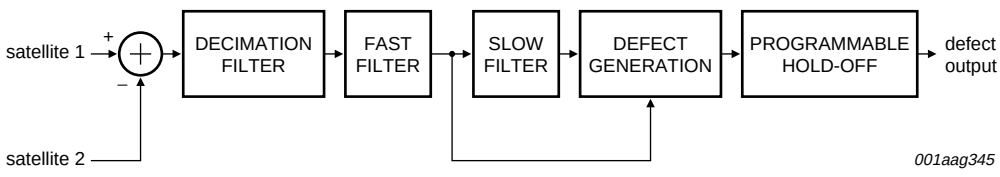


Fig 34. Defect detector block diagram

6.6.10 High level features

6.6.10.1 Automatic error handling

Three watchdogs are present:

- Focus: detects focus dropout of longer than 3 ms, sets focus lost interrupt, switches off radial and sledge servos, and disables drive to disc motor

- Radial play: started when radial servo is in On-track mode and a first subcode frame is found; detects when maximum time between two subcode frames exceeds the time set by the playwatchtime parameter; then sets radial error interrupt, switches radial and sledge servos off, and puts disc motor in jump mode
- Radial jump: active when radial servo is in long jump or short jump modes; detects when the off-track counter value decreases by less than four tracks between two readings (time interval set by jumpwatchtime parameter); then sets radial jump error, switches radial and sledge servos off to cancel jump

The focus watchdog is always active, the radial watchdogs are selectable via parameter radcontrol.

6.6.10.2 Automatic sequencers and timer interrupts

Two automatic sequencers are implemented (and must be initialized after power-on):

- Auto-start sequencer: controls the start-up of focus, radial and motor
- Auto-stop sequencer: brakes the disc and shuts down servos

When the automatic sequencers are not used it is possible to generate timer interrupts, defined by the time_parameter coefficient.

6.6.11 Driver interface

The control signals (pins RA, FO and SL) for the mechanism actuators are pulse-density modulated. The modulating frequency can be set to either 1.0584 MHz or 2.1168 MHz, controlled via parameter xtra_preset. An analog representation of the output signals can be achieved by connecting a 1st-order low-pass filter to the outputs.

During reset ($\overline{\text{RESET}}$ pin is held LOW) the RA, FO, and SL pins are high impedance. At all other times, when the laser is switched off, the RA and FO pins output a 2 MHz 50 % duty-cycle signal.

6.7 Flexi servo options

The Flexi servo contains some additional hardware:

- **LPF**: the low-pass filters construct a multi-bit representation of the incoming PDM stream arriving from the analog ADCs; the cut-off frequencies of all the filters are user-programmable from registers.
- **Fine DC-offset subtraction**: the fine DC-offset values are held in CD-Slim registers; these values can be subtracted from the LPF outputs.
- **Decimation filter**: The decimation filter behavior is controlled by the LPF cut-off frequency selection and passes only the n^{th} sample.
- **Interrupt generator**: this block raises an interrupt every time the output of the decimation filter becomes valid; the interrupt will either clear itself after a given time or can be cleared by the ARM microprocessor.
- **Servo registers**: registers that exist within the servo register address range; depending upon their function they will be either read-only or write/read registers; some of the flexible servo registers utilize the full 32-bits available to improve bandwidth performance for certain flexible servo operations.

- **Sigma-delta noise shaper**: this block regenerates a PDM data stream from a given multi-bit value, which is provided at its inputs.

6.7.1 Modes of operation

The flexible servo can be used in six main modes, they are described in the following sections.

6.7.1.1 Hardware servo-only

This mode uses the hardware servo (PDSIC) only without any additional processing taking place on either the diode input signals or the servo output signals.

6.7.1.2 Hardware servo with fine offset compensation

This mode is the same as hardware servo-only mode but has the addition of the fine offset compensation functionality in the digital domain. The fine offset compensation is additional to the coarse offset compensation, which is available in all flexible servo modes.

6.7.1.3 Fully flexible servo

Also known as software servo. The diode signals have the fine offset applied to them and are passed to registers for reading by the ARM microprocessor. The complete servo functionality is implemented in software running on the ARM; the PDSIC hardware servo is switched out of the loop entirely. The ARM calculates values, which are used to generate servo signals for driving the mechanism actuators. The PDM generation for the servo output signals is performed by the sigma-delta block.

6.7.1.4 Pre-processing with hardware servo

The diode signals have the fine offset applied to them and are passed to registers for reading by the ARM microprocessor. The ARM pre-processes these signals and they are fed back to the inputs of the hardware servo via sigma-delta noise shapers. The hardware servo (PDSIC) performs all the control functions (on the modified input signals) and outputs the servo signals as hardware servo-only mode.

6.7.1.5 Hardware servo with post-processing

The diode signals are passed directly to the hardware servo inputs (can be with, or without, fine-offset compensation added). The PDSIC servo output signals are passed to registers for reading by the ARM microprocessor. The ARM executes any post-processing it deems necessary on the signals and passes new values back which are used to drive the mechanism actuators. The servo outputs are therefore generated by the ARM, rather than the PDSIC, but based on the signals provided by the PDSIC.

6.7.1.6 Pre-processing with hardware servo plus post-processing

The diode signals have the fine offset applied to them and are passed to registers for reading by the ARM microprocessor. The ARM pre-processes these signals and they are fed back to the inputs of the hardware servo via sigma-delta noise shapers. The hardware servo performs all the control, and the servo output signals are passed to registers for reading by the ARM microprocessor. The ARM executes any post-processing it deems necessary on the signals and passes new values back which are used to drive the mechanism.

6.8 Block decoder

The general features of the SAF784x block decoder are:

- 32-bit microprocessor interface
- Channel decoder compatible C2I interface
- Segmentation manager compatible MiMeD2 interface

The main CD decode features of the SAF784x block decoder are:

- 16-bit data channel
- Data byte swapping
- Sync pattern (0x00, 0xFF to 0xFF, 0x00) detection and interpolation
- Main data de-scrambling (as per the *CD Yellow Book (ISO/IEC 10149)* for mode 1, mode 2 form 1 and mode 2 form 2 sectors)
- Header (MSF address) monitoring, interpolation and repair
- Firmware programmable 1-bit stream filtering on sector boundaries (CD-ROM only)
- Fast real-time C3 error correction (including automatic detection of mode 1, mode 2 form 1, and mode 2 form-2 sectors) using an internal two sector SRAM
- Separate 8-bit subcode channel
- Subcode P + Q channel de-interleaving and Q channel CRC checking
- Subcode CD-text mode four-packet extraction and CRC checking
- Automatic subcode stream filtering associated with the data stream filter
- Stream building (the main data stream, the block error byte, the error flag bytes and the subcode channel will be merged into a single data stream for transmission to the segment manager
- Status word tag creation (which contains sector-specific information)

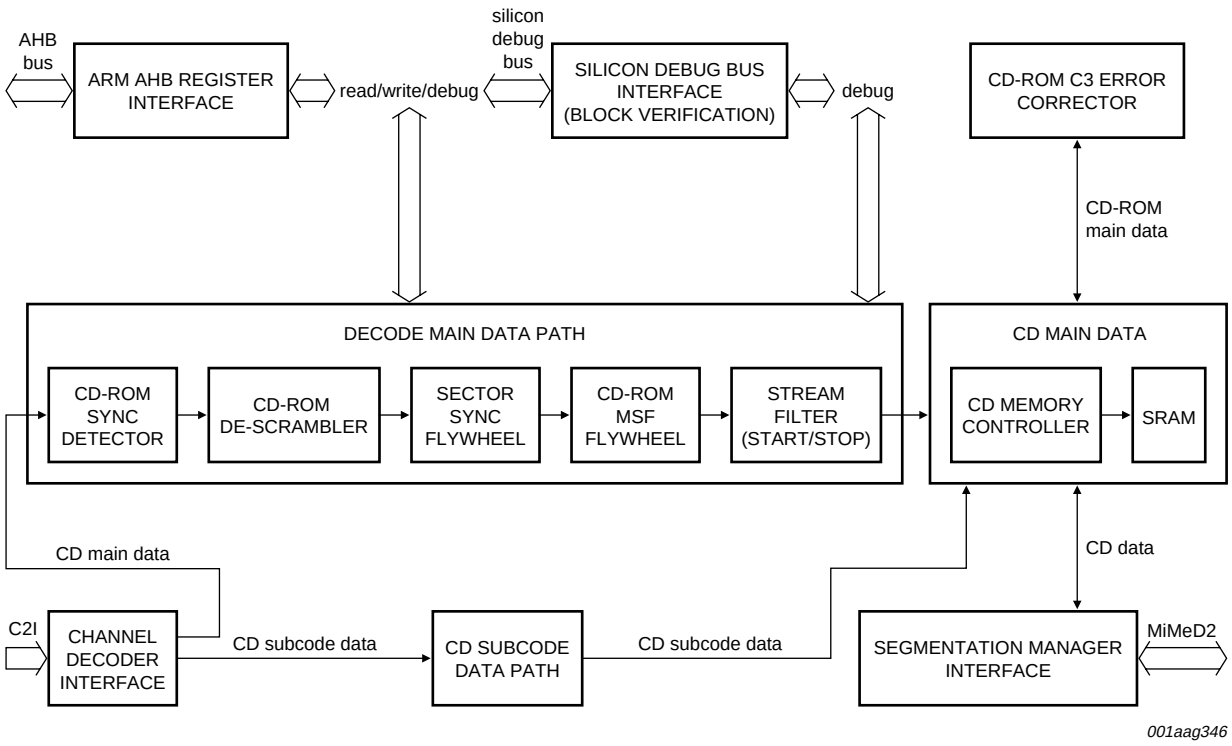


Fig 35. Block decoder data path

6.8.1 Supported modes of operation

The block decoder supports CD-DA and CD-ROM decode transfers.

CD main data and subcode data are received from CD-Slim over the C2I interface. The main data is processed by the decode main data path functions before being passed to the memory controller. In the memory controller the main data is assimilated with the subcode data and the C3 error corrector is run on CD-ROM main data. The sector data and some status information are then passed to the SAF784x segment manager over the MiMeD2 interface.

6.8.2 Channel decoder to block decoder interface (C2I)

The data interface between the channel decoder and the block decoder is an asynchronous interface; the channel decoder and the block decoder operate in independent clock domains.

Data is transferred over C2I in bursts of one EFM frame. Each EFM frame consists of one subcode data byte plus twelve 16-bit main data words with associated reliability flags.

CD-ROM data is word-aligned as per the *CD Yellow Book (ISO/IEC 10149)* with the first word of the CD-ROM sync pattern appearing on a left audio word. The subcode sync may, or may not, be aligned with the first word of the CD-ROM sync pattern depending on the alignment on the disc.

Subcode data is provided at the rate of one byte per EFM frame. Each byte contains a bit for each of the subcode channels, P-W (P channel is bit 7).

The alignment between the main channel and the subcode channel must be the same each time a sector is read.

6.8.3 Block decoder to segmentation manager interface

The MiMeD2 interface is the data interface between the block decoder and the SAF784x segment manager. It is an asynchronous interface; the block decoder and SAF784x segment manager operate in independent clock domains.

Data is transferred over MiMeD2 in bursts of one sector. The size and speed of the transfer is determined by the settings in the OP_CTRL register. Each transaction on MiMeD2 is accompanied by a toggle of the bld_req signal. The data and flags are updated on each transaction.

The transfer order is:

1. Main data: 1176 word16 (2352 bytes)
2. Flags data (optional): 148 word16 (296 bytes)
3. Subcode data (optional): 57 word16 (114 bytes)
4. Status words: two word16 (four bytes)

6.9 Segmentation manager

6.9.1 General

The segmentation manager controls the flow of block-decoded data from the block decoder and synchronous ARM system bus.

The segmentation manager consists of a 2 kB buffer, that is accessible on the ARM subsystem bus once the entire sector has been transferred from block decoder into the segmentation buffer.

The DMA cycle is then initiated to transfer sector data from the segmentation manager to the ARM processor memory for MP3 decoding to commence. The DMA transfers are expected to be continuous burst transfers to ARM processor memory and completed at the sector boundary.

The segmentation manager register SEL_WRITE_MODE allows access to the segmentation buffer either to the block decoder MiMeD interface or the synchronous ARM system bus.

Remark: Access to segmentation buffer from the MiMeD interface is Write access only, or Access only from synchronous ARM processor system bus is Read access only.

The segmentation buffer memory is 692-bit SRAM $\times$ 32-bit SRAM. The maximum number of 32-bit words per complete sector is 692 words.

6.9.2 Interrupt generation

An interrupt is generated on completion of transfer of every sector from the block decoder. The signal memory_full is used to generate an interrupt. The interrupt, once serviced by software, can be cleared by register write to Inreq_Clr.

Once the interrupt has been triggered, it will remain asserted until cleared by the Inreq_Clr signal. However, after being cleared, it is then free to fire again on the next transition.

Note that the register bit INREQ_CLR retains the latest value written by the ARM processor so a typical sequence will be to write logic 0 followed some time later by logic 1 in order to return to the non-reset state.

6.9.3 Segmentation buffer ARM sub-system interface

The segmentation buffer access to the ARM sub-system is fully synchronous. The implementation allows for minimum latency-overhead so as to maximize the available bandwidth on the ARM processor. The synchronous interface is AMBA AHB-compliant.

Note: The synchronous transfers are initiated by the Direct Memory Access (DMA) controller. Once sector transfer has been initiated, the recommendation is that these transfers should not be interrupted and the application must allow for the complete sector to be copied to the main ARM sub-system 110 kB internal memory.

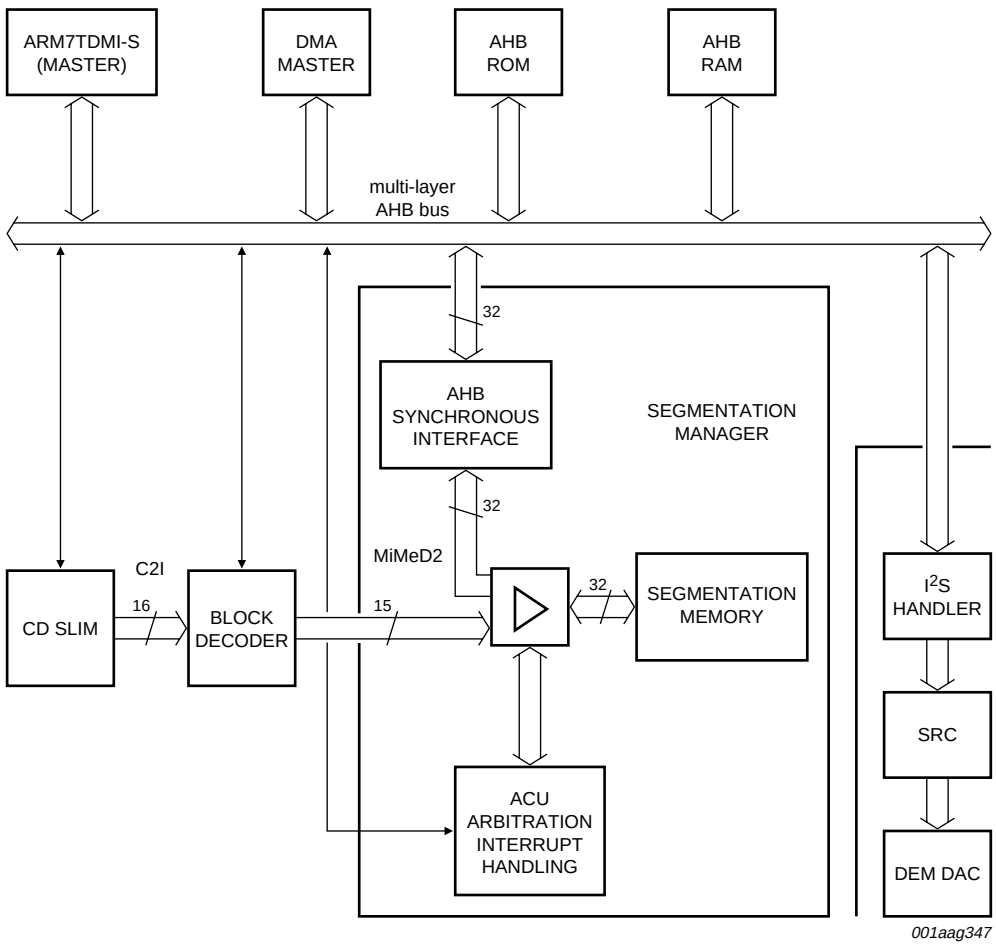


Fig 36. SAF784x data path with segmentation manager

6.10 Laser interface

The laser diode pre-amp function is built in to the SAF784x and is illustrated in Figure 37. The current, up to 120 mA, can be regulated in four steps ranging from 58 % up to full power. The voltage derived from the monitor diode is maintained at a steady state by the laser drive circuitry, regulating the current through the laser diode.

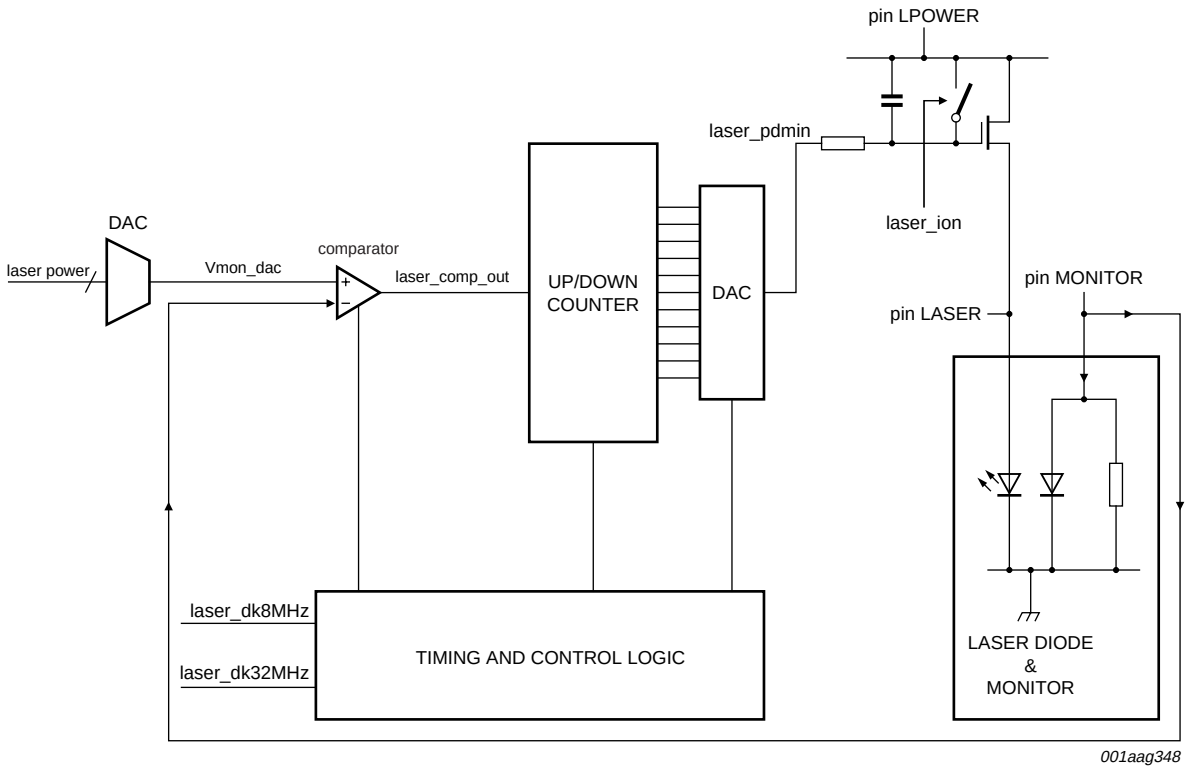


Fig 37. Block diagram of laser control circuit

7. ARM7 System

The following sections give a top-level description of the individual blocks.

7.1 ARM7TDMI-S microprocessor

The ARM7TDMI-S microprocessor is a member of the ARM family of 32-bit processors. The ARM processor offers a high performance with low power consumption and low gate count. The ARM architecture is based upon RISC. The RISC principles provide the following key benefits:

- High instruction throughput
- Excellent real-time interrupt response

Table 15. Performance characteristics for ARM7TDMI-S

Process technology	Performance (MIPS/MHz)	Power consumption (mW/MHz)	Maximum operating frequency (MHz)	Typical operating frequency requirements for SAF784x (MHz)
0.18 µm	0.9	0.39	76	76[1]

[1] The frequency of operation depends on the performance required for the SAA7834 application and the software complexity. MP3/WMA decoding requires most high-speed peripherals to operate at this frequency. The MP3/WMA decoding library is implemented in software.

The ARM7TDMI-S processor has two instruction sets:

1. The 32-bit ARM instruction set
2. The 16-bit ARM thumb instruction

The ARM uses a three-stage pipeline to increase the throughput of the flow of instructions to the processor. This enables several operations to operate simultaneously and the processor and memory systems to operate continuously.

The three-stage pipelines can be defined in the following stages:

- Fetch cycle: fetches the instruction from the memory
- Decode cycle: decodes the registers and the instructions fetched
- Execute cycle: fetches the data from register banks; the shift, ALU operations performed and data is written back to the memory

The microprocessors have traditionally the same width for the instructions and data. The 32-bit architecture would be more efficient in performance and could also address a much larger address space compared to 16-bit architectures. The code density for 16-bit architecture would be much higher than 32-bit and the performance would be greater than half the 32-bit performance.

The ARM thumb instructions concept addresses the issues when 16-bit instructions are used but the performance required is 32-bit architecture. Therefore the aim of the thumb instruction set can be summarized as follows:

- Higher performance for 16-bit architecture, if 16-bit instructions are to be used.
- The code density achieved with 16-bit instructions in a 32-bit architecture is the most efficient use of memory space.

7.2 Static Memory Interface Unit (SMIU)

The AHB SRAM controller implements an AHB slave interface to an external SRAM. This interface is only available in the development version of this device. The specification of this interface is described below:

- 32-bit AHB interface width
- 76 MHz maximum AHB operating frequency
- Configured for low latency
- Maximum of two SRAMs/ROMs/Flash/Burst ROM of 2 MB each, can be accessible
- 32-bit data

- AMBA AHB-compliant
- Asynchronous burst mode read access from burst mode ROM and Flash devices
- Asynchronous page mode read access in non-clocked memories
- 8-bit and 16-bit wide data paths
- Independent configuration of two memory banks, with maximum access of 2 MB each
- Programmable wait-state up to a maximum of 31. This parameter influences the response times from external memory, typically access times
- Programmable output-enable and write control-enable delays (15 cycles maximum)
- Byte lane select outputs for eight bits or 16 bits; this can be a useful feature if access is only required for either lower or upper bytes of data width
- Little endian configuration: this has been fixed
- Programmable chip-select polarity

7.2.1 SMIU operation modes

The static memory interface unit can connect to two external memories. Each of these two memories can be accessed sequentially. The static memory interface unit consist of memory banks that map the external memories to the main system memory address space. The versatility of the static memory interface unit enables access to a wide range of memory types with different memory access times.

The static memory interface unit only supports asynchronous memory types that do not require the use of a system clock.

Each of the two memory banks is capable of supporting the following memory types:

- SRAM/PSRAM
- ROM
- Flash EPROM
- Burst ROM memory

Each of the above memories can be configured to either 8-bit or 16-bit external memory data paths. The static memory interface unit has been configured to support only little endian operation.

7.2.2 Selecting the memory banks

The access to the memory begins by asserting the chip-select lines to each of the two memory banks supported.

The polarity of the chip-select lines can be programmed to be either active HIGH or active LOW (default).

A memory bank is selected by the ARM addressing the static memory interface unit. This is achieved on the SAF784x as follows:

- ARM AHB address[27:26] = 0, 0 selects bank 0
- ARM AHB address[27:26] = 0, 1 selects bank 1

7.2.3 Read access to external memories

The sections below describe the step-by-step process required for setting up the static memory interface unit registers prior to accessing the external memory.

7.2.3.1 Programming the external memory data widths

The bank configuration register SMBCRx (x = bank number, 0 or 1), is used to describe the bus width. The SAF784x can only support an external data bus width of 16-bits. If, during a data transfer requested by the internal ARM, the external data width is narrower than the ARM bus data width, several interface bus cycles may be required to complete the transfer. For example, if the external memory data bus is 16-bits wide, and a 32-bit read is requested by the ARM, the AHB bus is stalled until the static memory interface unit has fetched the two half words.

7.2.3.2 Wait-state generation

The wait-state assertion registers are SMBWST1Rx and SMBWST2Rx. Note that ‘x’ denotes the bank number (0 or 1).

The wait-state time is crucial, as the internal AHB bus cycles are scaled with respect to the access times of the external memory. The highest number of wait-states that can be asserted is 31 system clock cycles. The maximum AHB clock frequency for the SAF784x, is 76 MHz.

Hence the maximum access time that can be supported for this implementation of wait-state register is 13 ns $\times$ 31 cycles. The minimum wait-state assertion time is when the wait-state field in the register is 0 and the hardware asserts a wait time of 2 $\times$ 13 ns. This is the fastest external memory access time.

This is an important parameter to consider when selecting the memory type to interface with the static memory interface unit.

Each wait-state register has a particular importance for the following actions: SMBWST1Rx when performing read transfers from external memory; SMBWST2Rx when performing write transfers to external registers.

An example of the effect of wait-state assertion when reading from external memory is shown in Figure 38.

7.2.3.3 Output-enable delay programming

The output enable can be programmed, at the time it needs to be active. This is typically after the memory has been selected, by toggling the chip-select lines. The maximum programmable delay between when the chip-select is active and when the output-enable activates is 15 cycles. The output-enable delay value is programmed in field WSTOEN in register SMBCRx.

This feature is intended for memory that may not be able to provide valid output immediately after the chip-select lines are active. Note that the output-enable is de-asserted at the same time as the chip-select lines at the end of a transfer.

Remark: The output-enable delay programmed in WSTOEN must be less than the programmed wait-state. Note that the external memory access times are determined by the wait-states and not by the output-enable delay.

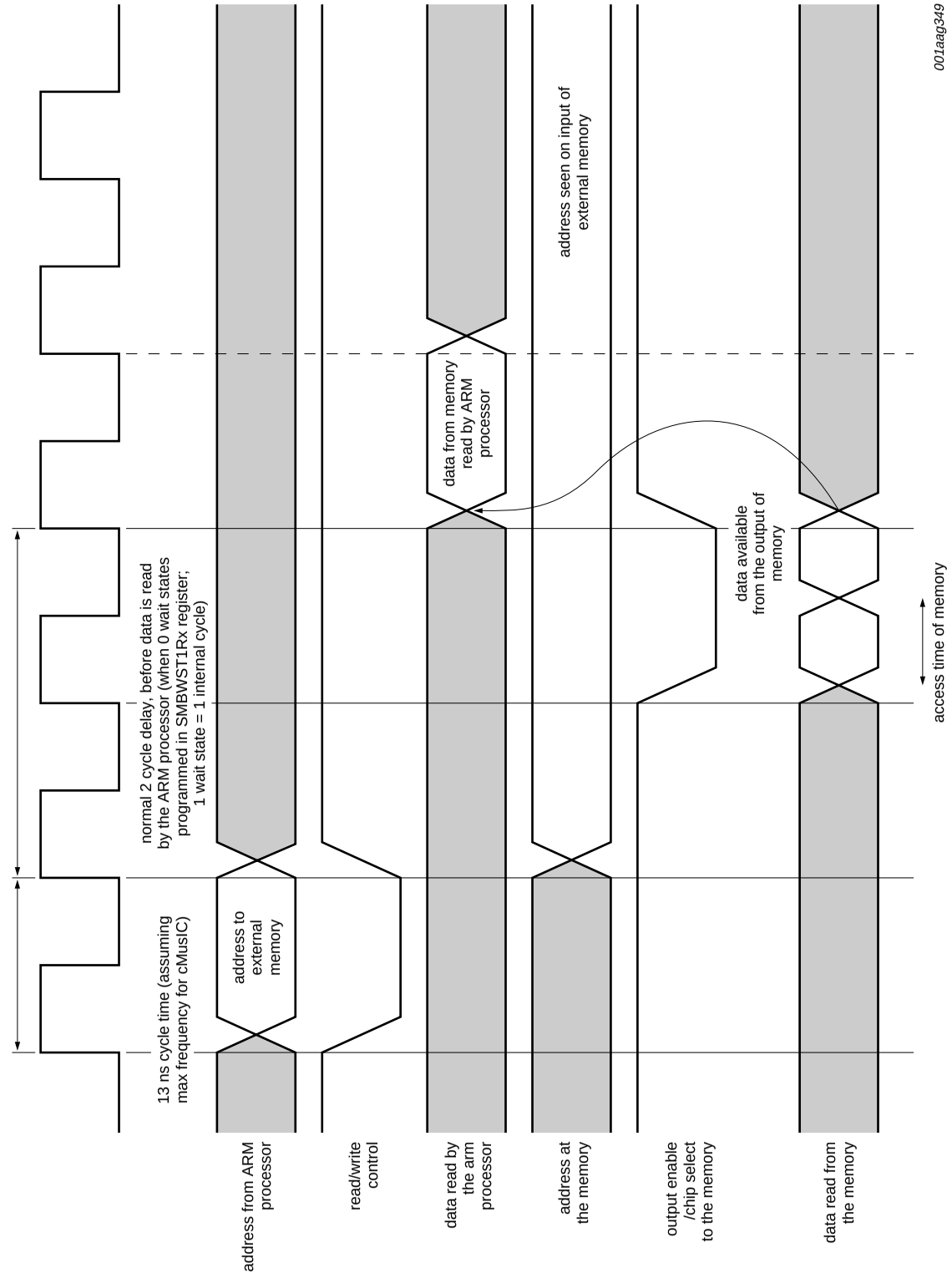


Fig 38. Default wait-state assertion during read from external memory

7.2.3.4 Burst reads from external memory

The static memory control unit can support a maximum of four consecutive reads from external reads. This feature supports burst mode ROM devices.

This feature increases the bandwidth for sequential reads compared to non-sequential reads. The burst access requires the user to specify the access times in SMBWST2x; for single reads, the access times are specified as wait-states in SMBWST1x.

The chip-select and output-enable lines are held during burst mode transfers.

It is also important to note that the burst transfers cannot cross quad boundaries. For eight bits, this implies: ARM address[1:0] = 11, and for 16 bits: ARM address[2:1] = 11.

Example for a 16-bit wide external memory:

1. If start address is ARM address[2:1] = 01, this address starts as a slow read, and hence the SMBWST1x value applies.
2. The next sequential address is ARM address[2:1] = 10 and 11. These addresses are fast reads and hence the SMBWST2x value applies.
3. The final address before the burst transfer completes is ARM address[2:1] = 00. The final read is slow (more wait-states apply), and the SMBWST1x value applies.

7.2.4 Write access to external memory

Writing to external memory requires a similar setup to that described for reading from external memory.

The write-enable parameter needs to be programmed.

Writing to external memory can be extended by applying wait-states as described for reading from external memory.

7.2.4.1 Write-enable programmable delay

This is the programmable delay between the asserted chip-select and the asserted write-enable. A total of 15 cycles can be asserted. The wait-state values are programmed in field WSTEN in register SMBCRx.

When no write-enable delay is programmed, the hardware introduces a default delay of one clock cycle between active chip-select and asserting write control.

The write-enable delay programmed in WSTEN must be less than the wait-state programmed in register SMBWST2x.

7.2.5 SMIU operation parameters to calculate latency

The delay through the logic to external memory are categorized in three sections:

- Default hardware latency, with no software-programmable increase in delay
- Software-programmable increase in latency delay
- Standard delay through logic including pads

The following requirements must be taken into account when computing the overall latency delay:

- Maximum ARM operating frequency for the SAF784x: for applications supporting WMA decoding, the ARM operating frequency is fixed at 76 MHz
- External memory access times: critical for determining the allowable delay that is to be programmed via software, or the delay through the hardware

The equation for read latency delay for sequential reads. This is the worst case latency, with no burst reads:

- Memory address delay to external memory = one internal clock cycle.
- External memory data to internal ARM = two internal cycle delay (fixed).
- Additional delay, to increased set-up times; maximum delay is given as follows:
  - two internal cycles (fixed by hardware) + 31 × cycle time (ARM processor frequency).
  - 31 cycles: the value that can be programmed in either register SMBWST1x or SMBWST2x.

7.3 Program ROM interface

The ROM interface provides an interface between the on-board 130 kB ROM memory and the ARM via the AHB bus. The interface specification is described below:

- 32-bit AHB interface width
- 76 MHz maximum AHB operating frequency
- Configured for low latency
- 32-bit data
- AMBA AHB-compliant

The low-latency architecture is optimized for low-speed operation. No wait-states are used and the ROM control signals are taken directly from the AHB bus. This means that the maximum frequency is likely to be limited by the speed at which the control signals arrive from the AHB master.

7.4 Boot ROM interface

The ROM interface provides an interface between the on-board 42 kB ROM memory and the ARM via the AHB bus. The interface specification is described below:

- 32-bit AHB interface width
- 76 MHz maximum AHB operating frequency
- Configured for low latency
- 32-bit data
- AMBA AHB-compliant

The low-latency architecture is optimized for low-speed operation. No wait-states are used and the ROM control signals are taken directly from the AHB bus. This means that the maximum frequency is likely to be limited by the speed at which the control signals arrive from the AHB master.

7.5 Embedded KFlash interface

The KFlash controller is an interface between the embedded Flash memory device and AHB bus.

The AHB embedded Flash controller connects embedded Flash memory devices to the AHB bus.

The embedded Flash controller supports full AHB bus protocol and will never generate a retry or split response.

The data path between the memory and the controller is fixed at 128-bit width to implement a double 128-bit cache line for creating a read performance comparable to reading SRAM, for cache hits.

The controller features a programmable number of wait cycles. A user can select between 0 and 255 wait cycles, to allow for an optimal performance with the chosen Flash instance at the clock frequency of the specific application.

The design is optimized to interface with an ARM CPU with embedded JTAG tap controller:

- 32-bit AMBA AHB protocol with 76 MHz AHB operating frequency
- AHB reads for sub-word and word size
- AHB register interface
- Zero wait-states, sustained read throughput on linear reads
- Programmable wait-state counter for read with cache miss, including zero wait-state for low frequencies
- Cache for 2 $\times$ 128-bit words
- JTAG interface access to Flash memory

7.6 RAM interface

The RAM interface provides an interface between the on-board 110 kB SRAM memory and the ARM via the AHB bus. The specification of this interface is described below.

- 32-bit AHB interface width
- 76 MHz maximum AHB operating frequency
- Configured for low latency
- AHB reads and writes for sub-words and word sizes
- 32-bit data

7.7 I^2C-bus interface

This interface can be used as an I^2C-bus slave or master and is fully compliant with the I^2C-bus specification. The specification of this interface is described as follows:

- Master/slave configurations
- Address 0x6E
- 76 MHz maximum AHB operating frequency

- 8.4672 MHz I^2C-bus operating frequency
- 4 B Rx FIFO depth
- 4 B Tx FIFO depth
- Maximum I^2C-bus frequency of 400 kHz
- Compatible with 7-bit and 10-bit addressing

7.8 General purpose I/Os

The GPIOs are linked to the VPB bus. This interface provides individual control over each bidirectional pin. Each pin can be configured to be an input, output or bidirectional:

- 32 bidirectional I/Os

7.9 Interrupt controller

- 26 dedicated internal interrupts
- Two external interrupts which have programmable polarity
- Two interrupt types available: Interrupt Request (IRQ), and Fast Interrupt Request (FIQ)
- Interrupts can be defined as IRQ or FIQ
- One of 32 priority levels can be assigned to an interrupt
- Interrupt priority threshold level
- All interrupts are maskable

7.10 UART interfaces

- Compatibility with UART industry standard 16550
- 16-deep transmit and receive FIFO size
- Receive FIFO with error flags
- Software-selectable baud rate generator including fractional pre-scaler
- Four selectable receive FIFO interrupt trigger levels
- Standard asynchronous error and framing bits (start, stop, and parity overrun, break)
- Maximum UART clock frequency of 50 MHz
- Transmit, Receive, Line Status, and Data Set interrupts independently controlled
- Fully-programmable character formatting
- Auto-baud functionality for detecting the incoming baud rate
- False start-bit detection (debounce)
- Complete status reporting capability
- Line break generation and detection
- Loop-back controls for isolating communication link faults
- Prioritized interrupt system controls
- Optional ARM DMA controller flow-control interface
- AMBA VPB-compliant register interface

7.11 Timers

- Conforms to the VLSI Peripheral Bus (VPB) interface specification
- Clock prescaler for external high-frequency sources
- The VPB timer operates in two fully independent clock domains:
  - vpb_clk for accessing control and status registers (76 MHz maximum)
  - timer_clk for the timer/counter function (50 MHz maximum)
- The VPB timer can support up to four match registers having:
  - Continuous operation with optional interrupt generation on match
  - Stop on match, with or without, interrupt generation
  - Reset on match, with or without, interrupt generation
- Optional external match notification pins with the following features:
  - Set LOW on match
  - Set HIGH on match
  - Toggle on match
  - No action on match
- Up to four capture registers and capture trigger pins with optional interrupt generation on a capture event
- Interrupt generation on match event and capture event

7.12 Watchdog timer

- Configurable watchdog feature including:
  - Watchdog timer restart trigger protection by key
  - Watchdog timer reload value protection by key access sequence
  - Watchdog timer reset disable (for debug) protection by key
- Interrupt generation watchdog time-out event

7.13 Real-time clock

- Zero wait-state to access all registers from the VPB interface
- Provide coherent fraction and seconds time
- Requirement for external 32 kHz crystal
- Alarm and tick interrupts will be generated even when the VPB clock is switched off
- VPB RTC interrupt
- Real-time clock and VPB clock may be asynchronous (the VPB clock frequency could be higher or lower than the RTC clock frequency)
- Maximum VPB slave frequency: 50 MHz

7.14 DMA controller

The Single operation DMA interface (SDMA), is a small AHB bus master specifically designed for bulk memory transfers over the ARM AHB bus.

For memory-to-memory transfers, the length of the operation is specified. When half of this length is reached, or when the end of the transfer has been reached, the CPU can be interrupted or the CPU can poll for notification of this event.

The SDMA controller has a maximum of six channels, each channel can be configured with its own source, destination, length and control information.

The SDMA controller is primarily dedicated from sector transfers from segmentation manager to the ARM sub-system RAM.

- Performs memory-to-memory copies in two AHB cycles, and memory-to-peripheral or peripheral-to-memory in three AHB cycles
- Supports byte, half-word and word transfers, and correctly aligns it over the AHB bus
- Compatible with ARM flow control, for single requests (sreq), last single requests (lsreq), terminal count info (tc) and DMA clearing (clr)
- SAF784x architecture supports little endian for data transfers
- Contains maskable interrupts for each raw IRQ

7.15 Back-end audio processing

The back-end audio processing entails the parallel-to-serial I^2S conversion, sample-rate conversion for MP3 decoding and EBU data format generation.

7.15.1 Parallel-to-serial I^2S conversion

- Can operate in both master and slave modes
- Capable of handling NXP I^2S format of 8-bit, 16-bit and 32-bit word sizes
- Mono and stereo audio data supported
- The sampling frequency can range (in practice) from 16 kHz to 48 kHz (16 kHz, 22.05 kHz, 32 kHz, 44.1 kHz or 48 kHz)
- Two FIFOs are provided as data buffers, one for transmitting and one for reception; the depth of these FIFOs is configurable in HDLi
- Generates interrupt request
- Generates two DMA requests
- Controls include reset, stop, and mute options
- DMA acknowledge signals

7.15.2 Variable sample-rate converter

The hardware sample-rate conversion receives inputs from a varying input source. The input is an I^2S stereo audio signal. The sample-rate conversion block converts the frequency into a fixed 44.1 kHz audio output signal. The block works at a fixed frequency: 16.9344 MHz (384×44.1 kHz, or 67.7376 MHz / 4).

The audio input frequencies can range from 8 kHz to 48 kHz. The block converts the I^2S input signal to a signal with a fixed sampling frequency of 44.1 kHz.

The incoming I^2S signal is stored in a buffer. The signal is upsampled by a variable upsampling factor N. After a variable hold, the signal is down-converted with a fixed down-sample factor M.

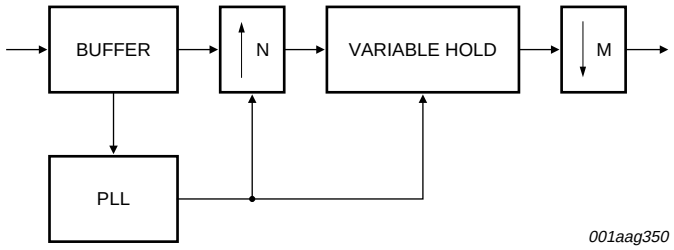


Fig 39. VSRC block diagram

7.15.3 EBU interface

The channel decoder contains a digital 1-wire EBU or S/PDIF output interface. It formats data according to specification IEC 958. The EBU rate can be selected to be 1× CD-speed or 2× CD-speed.

For proper operation of the EBU interface, the I^2S bclk must be internally generated, bit clock gating must be disabled, and the following relationship between ebuclk, I^2S bclk and I^2S-format must be true:

ebuclk = wclk × 64

Some fields in the user channel of the EBU stream can be filled by software.

7.16 Reset functionality

The device reset is crucial to the application as it allows the application to be initialized in the correct state before the key functions of the device can be exercised.

The device reset function depends on the following requirements:

- Power supply
- System clock
- Reset assertion

Normal reset application can be summarized as follows:

The device reset must be asserted asynchronously, that is, dependent only on the power supply and the reset pin $\overline{\text{RESET}}$, and de-asserted synchronously, that is, the system clock OSCIN needs to be available before starting the application.

7.16.1 Power supply requirements

The SAF784x is dependent on two supply voltages of 3.3 V and 1.8 V. It is recommended that the system solutions ensure the correct supply voltages are applied, during power-up and available during device initialization via a reset.

7.16.2 System clock

The SAF784x can support two types of external crystal frequencies of 8.4672 MHz and 16.9344 MHz. The digital logic is clocked at lower frequencies to ensure minimum power consumption during initialization.

The requirement for reset is the availability of the clocks during synchronous reset de-assertion.

7.16.3 Assertion of reset

The polarity of reset assertion is active LOW. During reset assertion the internal logic is initialized into the correct state. Due to the nature of complex logic, the initialization time may not be instantaneous. The minimum time that needs to be adhered to for the device to initialize into its correct state is shown in Figure 40.

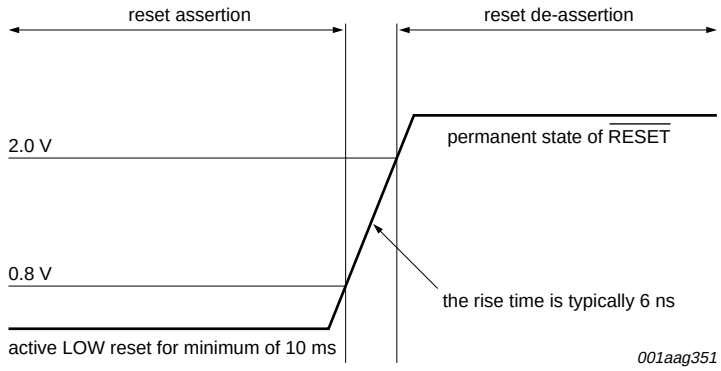


Fig 40. 5 V tolerant $\overline{\text{RESET}}$ input

During reset de-assertion, the main criterion is to ensure that the fully-synchronous internal logic is supplied with internal system clocks.

8. Limiting values

Table 16. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
$V_{DDD(C)}$	core digital supply voltage			–0.5	+2.5	V
V_{DDD}	digital supply voltage			–0.5	+3.6	V
V_{DDA}	analog supply voltage			–0.5	+3.6	V
$V_{I(a)}$	analog input voltage			–0.5	V_{DDA} + 0.5	V
$V_{I(dig)}$	digital input voltage		[1]	–0.5	+5.5	V
V_{esd}	electrostatic discharge voltage	human body model		2000	-	V
		machine model		200	-	V
T_{stg}	storage temperature			–55	+125	°C

[1] All digital inputs and bidirectional pins are 5 V tolerant.

9. Recommended operating conditions

Table 17. Operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{DDD(C)}$	core digital supply voltage		1.65	1.80	1.95	V
V_{DDD}	digital supply voltage		3.0	3.3	3.6	V
V_{DDA}	analog supply voltage		3.0	3.3	3.6	V
T_{amb}	ambient temperature		–40	+25	+85	°C

10. Characteristics

Table 18. Characteristics

V_{DDD} = V_{DDA} = 3.0 V to 3.6 V; $V_{DDD(C)}$ = 1.65 V to 1.95 V; T_{amb} = 25 °C; unless otherwise stated.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Supply (T_{amb} = –40 °C to +85 °C)							
$V_{DDD(C)}$	core digital supply voltage			1.65	1.8	1.95	V
V_{DDD}	digital supply voltage			3.0	3.3	3.6	V
V_{DDA}	analog supply voltage			3.0	3.3	3.6	V
$I_{DDD(C)}$	core digital supply current	$V_{DDD(C)}$ = 1.8 V	[1]	9.6[2]	15.5	42.4	mA
I_{DDD}	digital supply current	V_{DDD} = 3.3 V	[1]	11.1[2]	11.1	11.1	mA
I_{DDA}	analog supply current	V_{DDA} = 3.3 V	[3]	63.6[2]	63.6	63.6	mA
Analog section (V_{DDA} = 3.3 V; V_{SSA1} = V_{SSA2} = V_{SSA3} = 0 V)							
LF path							
Inputs: R1 and R2							
V_M	peak voltage	unipolar		20	-	960	mV
		bipolar		± 20	-	± 960	mV
ΔG/G	gain variation	within one channel		–20	-	+20	%
		between two channels		–3	+0	+3	%
V_{canDC}/V_{range}	DC cancellation voltage to range voltage ratio	unipolar		-	± 66	-	
		bipolar		-	± 33	-	
$V_{canDCacc}/V_{range}$	accuracy of DC cancellation voltage to range voltage ratio	full scale		-	± 4.1	-	%
f_s	sampling frequency			-	4.2336	-	MHz
f_i	input frequency			-	8.4672	-	MHz
B	bandwidth	recovered		20	-	-	kHz
S/N	signal-to-noise ratio	0 kHz to 20 kHz		55	-	-	dB
THD	total harmonic distortion	0 kHz to 20 kHz		-	-	–30	dB
R_i	input resistance	B = 0 kHz to 20 kHz		20	-	-	kΩ
$\Delta R_i/R_i$	relative input resistance variation			–30	-	+30	%
$V_{I(cm)}$	common-mode input voltage			-	1.6	-	V
V_{offset}	offset voltage	relative to OPU_REF_OUT		–30	-	+30	mV
HF path							
Inputs: D1, D2, D3 and D4							

Table 18. Characteristics *...continued*

V_{DDD} = V_{DDA} = 3.0 V to 3.6 V; $V_{DDD(C)}$ = 1.65 V to 1.95 V; T_{amb} = 25 °C; unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$V_{I(dif)(p-p)}$	peak-to-peak differential input voltage	6-bit ADC	1.4	1.4	1.4	V
$V_{I(cm)}$	common-mode input voltage	6-bit ADC	2	-	-	V
B	bandwidth	up to 6× (2 MHz × X-rate)	12	-	-	MHz
$\Delta t_{d(\varphi)}$	phase delay time variation	up to 6× (10 ns / X-rate)	-	-	1.66	ns
S/N	signal-to-noise ratio	100 Hz to 12 MHz	-	-	28	dB
$V_{i(ADC)se(p-p)}$	peak-to-peak single-ended ADC input voltage	at 6 MHz peak-to-peak	-	-	1	V
THD	total harmonic distortion	at 6 MHz	-	-	–35	dB
PSRR	power supply rejection ratio		40	-	-	dB
G_{tot}	total gain		2.4	-	38.4	dB
Z_i	input impedance	nominal	20	20	20	kΩ
B	bandwidth	–3 dB point	27	-	46	MHz
Audio DAC						
Input/Output: DAC_VREF outputs: DAC_LN, DAC_LP, DAC_RN and DAC_RP						
S/N	signal-to-noise ratio	A-weighted	-	90	-	dB
THD	total harmonic distortion	at 1 kHz	-	-	–80	dB
Audio feature						
Inputs: AUX_L and AUX_R						
S/N	signal-to-noise ratio	referenced to LF path values	-	60	65	dB
THD	total harmonic distortion	referenced to LF path values	-	–60	–30	dB
Laser driver						
Input: Monitor						
$I_{o(MONITOR)}$	output current on pin MONITOR		120	-	-	mA
$t_{startup(laser)}$	laser start-up time		1	-	-	ms
$V_{n(MONITOR)}$	noise voltage on pin MONITOR		–1	-	+1	mV
$V_{MONITOR(DC)}$	DC voltage on pin MONITOR	sel180 = 0	145	-	155	mV
		sel180 = 1	175	-	185	mV
IREF reference						
Output: OPU_REF_OUT						
$V_{ref(bg)}$	band gap reference voltage		1.14	1.2	1.26	V
I_O	output current		20	25	30	µA
Oscillator						
Pin: OSCIN (external clock)						
V_I	input voltage		-	$0.5V_{DDA}$	-	V
t_{iH}/T	relative HIGH input time		45	-	55	%
I_{LI}	input leakage current		–20	-	+20	µA
C_i	input capacitance		-	-	7	pF

Table 18. Characteristics *...continued*

V_{DDD} = V_{DDA} = 3.0 V to 3.6 V; $V_{DDD(C)}$ = 1.65 V to 1.95 V; T_{amb} = 25 °C; unless otherwise stated.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Pin: OSCOUT							
f_{osc}	oscillator frequency	crystal	[4]	8.4672	-	16.9344	MHz
		resonator		8.4672	-	16.9344	MHz
g_m	transconductance			17	-	-	mS
C_{fbck}	feedback capacitance			-	-	2	pF
C_o	output capacitance			-	-	7	pF
$R_{bias(int)}$	internal bias resistance			-	200	-	kΩ
Real time clock oscillator							
Pin: OSC_32K_IN (external clock)							
V_{IL}	LOW-level input voltage			-	-	$0.2V_{DDA}$	V
V_{IH}	HIGH-level input voltage			$0.8V_{DDA}$	-	-	V
t_{iH}/T	relative HIGH input time	relative to period		45	-	55	%
I_{LI}	input leakage current			-	1.5	2.5	µA
C_i	input capacitance			-	-	7	pF
Pin: OSC_32K_OUT							
f_{osc}	oscillator frequency	crystal	[4]	-	32.768	-	kHz
		resonator		-	32.768	-	kHz
g_m	transconductance			-	4	-	mS
C_o	output capacitance			-	100	300	pF
Pinning characteristics (T_{amb} = −40 °C to +85 °C)							
General							
I_{IL}	LOW-level input current	V_i = 0 V; no pull up		-	-	1	µA
I_{IH}	HIGH-level input current	$V_i = V_{DDD}$		-	-	1	µA
I_{OZL}	LOW off-state output current	V_O = 0 V or $V_O = V_{DDD}$		-	-	1	µA
I_{latch}	I/O latch-up current	$-(0.5V_{DDD}) < V_I < (1.5V_{DDD})$; T_j < 125 °C		100	-	-	mA
Power							
I_{cont}	continuous current			-	-	98	mA
Digital pins (T_{amb} = −40 °C to +85 °C)							
DC specifications; input							
V_{IH}	HIGH-level input voltage			2.0	-	-	V
V_{IL}	LOW-level input voltage			-	-	0.8	V
V_{hys}	hysteresis voltage			0.4	-	-	V
DC specifications; output							
V_{OH}	HIGH-level output voltage			V_{DDD} − 0.4	-	-	V
V_{OL}	LOW-level output voltage			-	-	0.4	V

Table 18. Characteristics *...continued*

V_{DDD} = V_{DDA} = 3.0 V to 3.6 V; $V_{DDD(C)}$ = 1.65 V to 1.95 V; T_{amb} = 25 °C; unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{OH}	HIGH-level output current	5 ns slew rate output; $V_{OH} = V_{DDD} - 0.4$ V	–5	-	-	mA
		12 mA output; $V_{OH} = V_{DDD} - 0.4$ V	–13	-	-	mA
		27 mA output; $V_{OH} = V_{DDD} - 0.4$ V	–28	-	-	mA
I_{OL}	LOW-level output current	5 ns slew rate output; V_{OL} = 0.4 V	4	-	-	mA
		12 mA output; V_{OL} = 0.4 V	11	-	-	mA
		27 mA output; V_{OL} = 0.4 V	27	-	-	mA
I_{OSH}	HIGH-level short-circuit output current	V_{OH} = 0 V	-	-	–45	mA
I_{OSL}	LOW-level short-circuit output current	$V_{OL} = V_{DDD}$	-	-	50	mA
I_{pd}	pull-down current	$V_I = V_{DDD}$	20	50	75	µA
		V_I = 5 V	20	50	75	µA
I_{pu}	pull-up current	V_I = 0 V	–13	–50	–40	µA
		$V_{DDD} < V_I < 5.0$ V	0	0	0	µA
AC specifications; input						
t_r	rise time		-	6	200	ns
t_f	fall time		-	6	200	ns
AC specifications; output						
t_{THL}	HIGH to LOW transition time	load = 30 pF; transition time read at 10 % and 90 % of output slope				
		5 ns slew rate output	-	4.0	-	ns
		12 mA output	-	2.9	-	ns
		27 mA output	-	3.8	-	ns
t_{TLH}	LOW to HIGH transition time	load = 30 pF; transition time read at 10 % and 90 % of output slope				
		5 ns slew rate output	-	4.0	-	ns
		12 mA output	-	2.9	-	ns
		27 mA output	-	3.8	-	ns

[1] Pins VDDD1, VDDD2 and VDDD3.

[2] Initial reset value with primary clock = 76 MHz, AHB and decoder = 4 MHz.

[3] Pins VDDA1, VDDA2 and VDDA3.

[4] It is recommended that the nominal running series resistance of the crystal or ceramic resonator is ≤ 60 Ω.

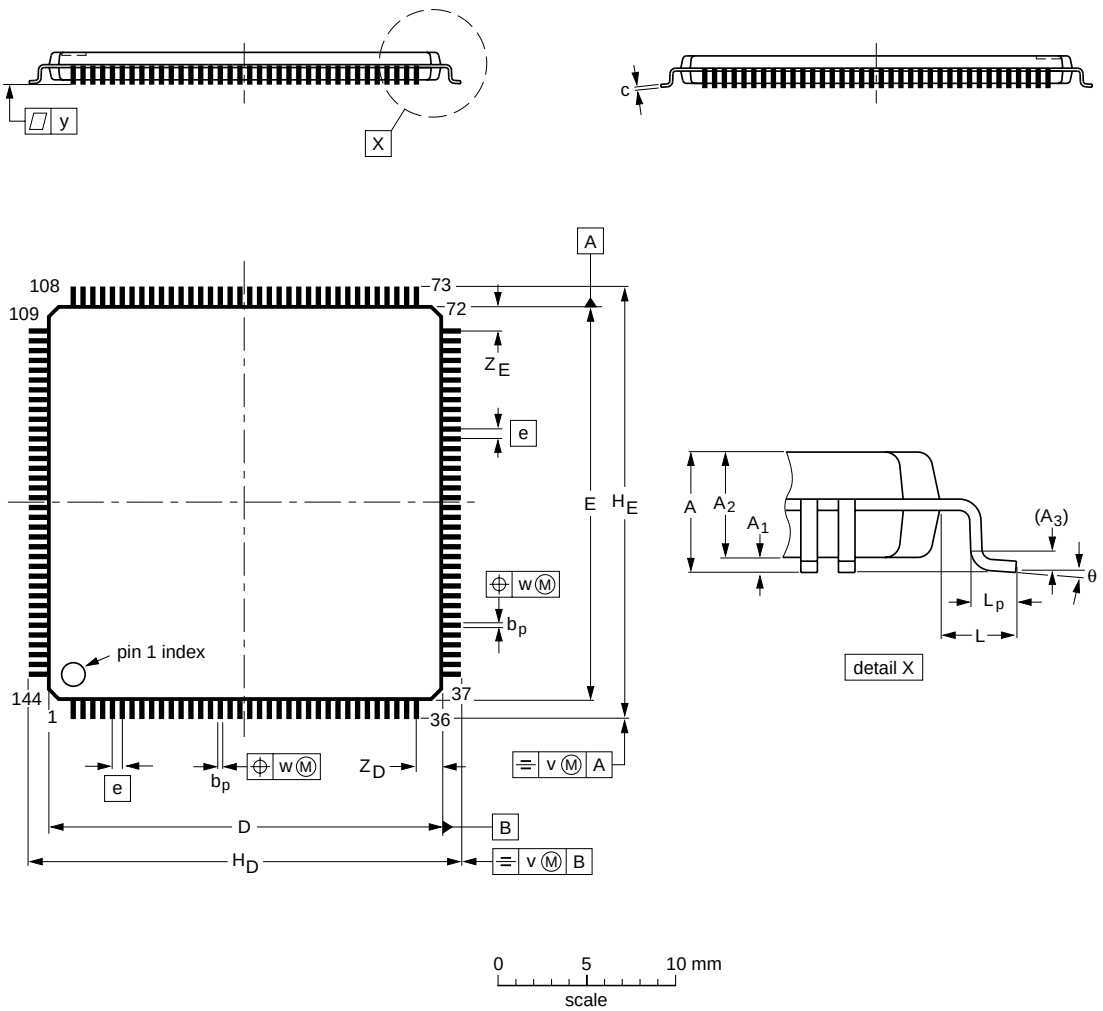
11. Test information

11.1 Quality information

This product has been qualified in accordance with the Automotive Electronics Council (AEC) standard *Q100 - Stress test qualification for integrated circuits*, and is suitable for use in automotive applications.

12. Package outline

LQFP144: plastic low profile quad flat package; 144 leads; body 20 x 20 x 1.4 mm **SOT486-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_D	H_E	L	L_p	v	w	y	$Z_D^{(1)}$	$Z_E^{(1)}$	θ
mm	1.6	0.15 0.05	1.45 1.35	0.25	0.27 0.17	0.20 0.09	20.1 19.9	20.1 19.9	0.5	22.15 21.85	22.15 21.85	1	0.75 0.45	0.2	0.08	0.08	1.4 1.1	1.4 1.1	7^o 0^o

Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT486-1	136E23	MS-026				~~00-03-14~~ 03-02-20

Fig 41. Package outline SOT486-1 (LQFP144)

13. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 "Surface mount reflow soldering description"*.

13.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

13.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

13.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

13.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 42) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 19 and 20

Table 19. SnPb eutectic process (from J-STD-020C)

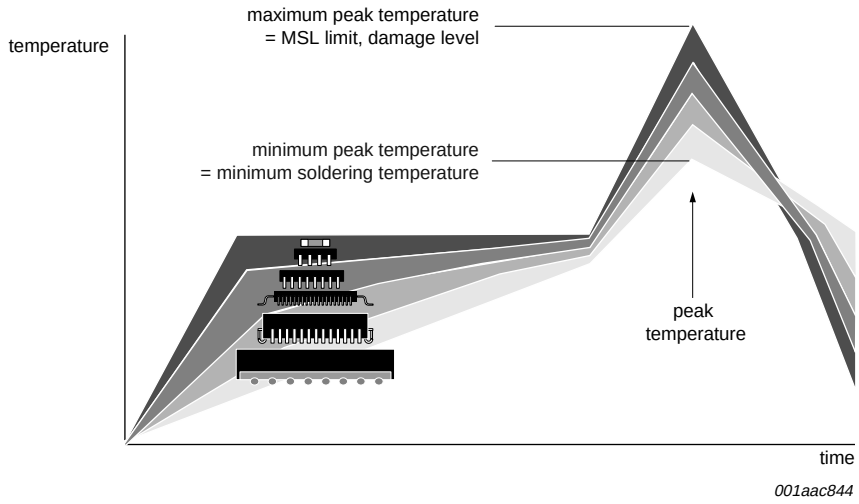
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm^3)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 20. Lead-free process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm^3)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 42.



MSL: Moisture Sensitivity Level

Fig 42. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

14. Abbreviations

Table 21. Abbreviations

Acronym	Description
ACU	Address Calculation Unit
ADC	Analog-to-Digital Converter
AGC	Automatic Gain Control
AHB	ARM Advanced High Performance Bus
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
AOC	Automatic Offset Compensation
ARM	Advanced RISC Machines (32-bit microprocessor design)
BLER	Block Error Rate
Cn	Check bit n
CA	Central Aperture
CAV	Constant Angular Velocity
CD	Compact Disc
CD-DA	Compact Disc Digital Audio
CD-MP3	Compact Disc Moving Picture Experts Group
CD-R	Compact Disc Recordable
CD-ROM	Compact Disc Read-Only Memory
CD-RW	Compact Disc Recordable/Writable
CD-WMA	Compact Disc Windows Media Audio

Table 21. Abbreviations *...continued*

Acronym	Description
CIRC	Cross Interleave Reed-Solomon Code
CLV	Constant Linear Velocity
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DAC	Digital to Analog Converter
DEM DAC	Dynamic Element Matching DAC
DC	Direct Current
DMA	Direct Memory Access
EFM	Eight to Fourteen Modulation (EFM+ has added bits for CD coding with no digital content)
ERCO	Error Corrector
FIFO	First in First Out
GPAI	General Purpose Analog Input
GPIO	General Purpose Input Output
HDLi	High-level Description Language integrator
HF	High Frequency
HPF	High Pass Filter
HSI	Hardware Software Interface
ICE	In-Circuit Emulator
IIR	Infinite Impulse Response
IRQ	Interrupt ReQuest
LF	Low Frequency
LPF	Low Pass Filter
LSB	Least Significant Bit
MiMeD2	Minimum Memory Decoder interface Decoder second generation
MP3	Moving Picture Experts Group
MSB	Most Significant Bit
MSF	Minutes Seconds Frames
NF	Noise Filter
OPU	Optical Pick Up
PDM	Pulse Density Modulation
PDSIC	Parallel Digital Servo Integrated Circuit
PI	Proportional-Integral
PID	Proportional-Integral and Differential
PLL	Phase-Locked Loop
PSRAM	Pseudo-Static Random Access Memory
PWM	Pulse-Width Modulator
RAM	Random Access Memory
RE	Radial Error
RISC	Reduced Instruction Set Computer
RL	Run Length

Table 21. Abbreviations *...continued*

Acronym	Description
ROM	Read Only Memory
RTC	Real Time Clock
SACD	Super Audio Compact Disc
SMIU	Static Memory Interface Unit
S/PDIF	Sony/Philips Digital Interface Format
SRAM	Static Random Access Memory
SRC	Sample Rate Converter
TPI	Track Position Indicator
UART	Universal Asynchronous Receiver Transmitter
VLSI	Very Large Scale Integration
VPB	VLSI Peripheral Bus
VSRC	Variable Sample Rate Converter
WDT	WatchDog Timer
WMA	Windows Media Audio

15. Glossary

ARM7TDMI-S — Specific version of ARM microprocessor used in the SAF784x (ARM7 family)

Dark currents — Currents caused by unwanted light leakage into the OPU causing offsets, otherwise known as dark current offsets

Flexi servo — Hardware which gives the ARM microprocessor access to the servo input signals and to drive the servo outputs. Allows servo algorithms to be performed in software in the ARM core.

I^2C — Inter IC Communication format

I^2S — Inter IC Sound format

PDSIC — Parallel Digital Servo IC (digital servo block within SAF784x)

Thumb — ARM 16-bit instruction set

16. Revision history

Table 22. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
SAF784X_2	20080509	Product data sheet	-	SAF784X_1
Modifications:	• Table 18 “Characteristics”: clarified presentation of ambient temperature information • Replaced instances of <tbd> with text or values in Section 6.5.10.7, Table 16 and Table 18.			
SAF784X_1	20071214	Preliminary data sheet	-	-

17. Legal information

17.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

17.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

17.3 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) may cause permanent damage to the device. Limiting values are stress ratings only and operation of the device at these or any other conditions above those given in the Characteristics sections of this document is not implied. Exposure to limiting values for extended periods may affect device reliability.

Terms and conditions of sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, including those pertaining to warranty, intellectual property rights infringement and limitation of liability, unless explicitly otherwise agreed to in writing by NXP Semiconductors. In case of any inconsistency or conflict between information in this document and such terms and conditions, the latter will prevail.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

17.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I^2C-bus — logo is a trademark of NXP B.V.

18. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

19. Contents

continued >>

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 9 May 2008
Document identifier: SAF784X_2