Technical Documentation

IIR 4.x - API Definition

Document Number:	L1D_AS411-1
TI Department	European Wireless Terminal Chipset Business Unit
Version:	1.0
Status:	APPROVED
Date:	March 16, 2006

Copyright © 2003 Texas Instruments
Texas Instruments – Proprietary Information
Strictly Private

Important Notice

IMPORTANT NOTICE

Texas Instruments Incorporated and / or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and / or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and / or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and / or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and / or software may be based on or implements industry recognized standards and that certain third party may claim intellectual property rights therein. The supply of products and / or the licensing of software do not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third–party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of TI.

Change History

Date of change	Changed by	Approved by	Approval date	Version	Notes
June 07, 2005	Thierry Le Gall			0.1	(1)
June 22, 2005	Thierry Le Gall			0.2	(2)
June 22, 2005	Thierry Le Gall	Sébastien Guiriec	June 22, 2005	1.0	(3)

Notes

(1) Creation of the document.
(2) Changes: API parameters ordering changed (2.2). Added mention to reset function (2.1.3). Updated A. Appendix. Added B. Appendix.
(3) Approval.

Table of Contents

List of Figures and Tables

Glossary

API	Application Protocol Interface
IIR	Infinite Impulse Response
FIR	Finite Impulse Response
SOS	Second Order Section

References

[1] L1D_AS410 – IIR 4.x – Overview

1 Introduction

The purpose of this document is to describe the Application Protocol Interface (API) related to the Equalization Filter (IIR) [1]. This document applies for IIR 4.0 and next upgrades IIR 4.x.

First chapter describes the API of the IIR 4.x module. A. Appendix presents the API summary with the functional block diagram of the IIR 4.x module.

2 IIR 4.x Module API

This chapter describes the parameter interface of the IIR 4.x module.

2.1 Entry Functions

2.1.1 Function f_iir4x_top ()

Prototype:

```
void      f_iir4x_top (T_IIR4X_STATIC_VAR *p_iir4x_data);
```

Description:

This IIR top level function contains the call of all signal processing functions necessary to perform the IIR algorithm. The function arguments are presented below (Table 2.1).

Arguments:

Type	Name	Flow	Description
T_IIR4X_STATIC_VAR	*p_iir4x_data	IN/OUT	Pointer on the IIR module static variables

Table 2.1 The IIR 4.x Top Function Arguments

The T_IIR4X_STATIC_VAR *p_iir4x_data structure pointer parameter is used to pass the static variables to the IIR top function as well as through the internal signal processing functions. In addition, the IIR configuration parameters are passed to the module through a data structure pointer T_IIR4X_PARAM *p_iir4x_param element of *p_iir4x_data.

Requirements:

The entire signal processing functions code must be mapped on a single DSP page as it does not support extended addressing.

Reentrancy:

This API is reentrant.

Return value:

None.

2.1.2 Function f_iir4x_init ()

Prototype:

```
void      f_iir4x_init (T_IIR4X_STATIC_VAR *p_iir4x_data);
```

Description:

This function is used to initialize the IIR module. The function arguments are presented below (Table 2.2)

Arguments:

Type	Name	Flow	Description
T_IIR4X_STATIC_VAR	*p_iir4x_data	IN/OUT	Pointer on the IIR module static variables

Table 2.2 The IIR 4.x Initialization Function Arguments

Requirements:

The init function code must be mapped on a single DSP page as it does not support extended addressing.

Reentrancy:

This API is reentrant.

Return value:

None.

2.1.3 Function f_iir4x_reset ()

Prototype:

```
void    f_iir4x_reset (T_IIR4X_STATIC_VAR *p_iir4x_data);
```

Description:

This function is used to reset the IIR module (FIR and SOS memory buffers). This function could be used during handover. The function arguments are presented below (Table 2.2)

Arguments:

Type	Name	Flow	Description
T_IIR4X_STATIC_VAR	*p_iir4x_data	IN/OUT	Pointer on the IIR module static variables

Table 2.3 The IIR 4.x Reset Function Arguments

Requirements:

The init function code must be mapped on a single DSP page as it does not support extended addressing.

Reentrancy:

This API is reentrant.

Return value:

None.

2.2 IIR 4.x Module Interface

This chapter focuses on the API for IIR module. Size of IIR 4.x API is 97 words. The API description is presented below (Table 2.4, Table 2.5, Table 2.6 and Table 2.7). The relation between API and functional block diagram is presented in (A. Appendix). The format and the range of API parameters are presented below (Table 2.8).

&	Type	Variable Name	Description
+00	T_UINT16	d_iir4x_control	Equalization enable/disable
+01	T_UINT16	d_iir4x_frame_size	Equalization IO frame size
+02	T_UINT16	d_iir4x_fir_swap	FIR/IIR structures swap
+03	T_IIR4X_FIR_PARAM	d_iir4x_fir_filter	FIR parameters
+46	T_IIR4X_SOS_PARAM	d_iir4x_sos_filter	IIR (SOS) parameters
+96	T_SINT16	d_iir4x_gain	Digital gain after equalization (D)

Table 2.4 The IIR 4.x API Parameters Description

&	Type	Variable Name	Description
+03	T_UINT16	d_iir4x_fir_filter.d_iir4x_fir_enable	0: NO FIR, 1: FIR Enabled
+04	T_UINT16	d_iir4x_fir_filter.d_iir4x_fir_length	FIR length N (max = 40)
+05	T_SINT16	d_iir4x_fir_filter.d_iir4x_fir_shift	FIR output scaling down
+06	T_SINT16	d_iir4x_fir_filter.a_iir4x_fir_taps[40]	FIR taps h0,..h39

Table 2.5 The IIR 4.x API Parameters – FIR Part

&	Type	Variable Name	Description
+46	T_UINT16	d_iir4x_sos_filter .d_iir4x_sos_enable	0: NO IIR, 1: IIR Enabled (SOS)
+47	T_UINT16	d_iir4x_sos_filter .d_iir4x_sos_number	IIR number of biquads (max = 6)
+48	T_IIR4X_ SINGLE_SOS_PARAM	d_iir4x_sos_filter .a_iir4x_sos_filter[6]	Biquads parameters

Table 2.6 The IIR 4.x API Parameters – IIR Part

&	Type	Variable Name	Description
+48	T_SINT16	&a_iir4x_sos_filter[1]->a_iir4x_sos_fact	SOS 1, scaling factors G_1'
+49	T_SINT16	&a_iir4x_sos_filter[1]->a_iir4x_sos_fact_form	SOS 1, Q-formats of G_1’
+50	T_SINT16	&a_iir4x_sos_filter[1]->a_iir4x_sos_den[2]	SOS 1 denominator: $b_{1,1}$, $b_{2,1}$
+52	T_SINT16	&a_iir4x_sos_filter[1]->a_iir4x_sos_num[3]	SOS 1, numerator: $a_{0,1}$ $a_{1,1}$, $a_{2,1}$
+55	T_SINT16	&a_iir4x_sos_filter[1]->a_iir4x_sos_num_form	SOS 1, Q-format of $a_{0,1}$ $a_{1,1}$, $a_{2,1}$
+56	T_SINT16	&a_iir4x_sos_filter[2]->a_iir4x_sos_fact	SOS 2, scaling factors G_2'
+57	T_SINT16	&a_iir4x_sos_filter[2]->a_iir4x_sos_fact_form	SOS 2, Q-formats of G_2’
+58	T_SINT16	&a_iir4x_sos_filter[2]->a_iir4x_sos_den[2]	SOS 2, denominator: $b_{1,2}$, $b_{2,2}$
+60	T_SINT16	&a_iir4x_sos_filter[2]->a_iir4x_sos_num[3]	SOS 2, numerator: $a_{0,2}$ $a_{1,2}$, $a_{2,2}$
+63	T_SINT16	&a_iir4x_sos_filter[2]->a_iir4x_sos_num_form	SOS 2, Q-format of $a_{0,2}$ $a_{1,2}$, $a_{2,2}$
+64	T_SINT16	&a_iir4x_sos_filter[3]->a_iir4x_sos_fact	SOS 3, scaling factors G_3'
+65	T_SINT16	&a_iir4x_sos_filter[3]->a_iir4x_sos_fact_form	SOS 3, Q-formats of G_3’
+66	T_SINT16	&a_iir4x_sos_filter[3]->a_iir4x_sos_den[2]	SOS 3, denominator: $b_{1,3}$, $b_{2,3}$
+68	T_SINT16	&a_iir4x_sos_filter[3]->a_iir4x_sos_num[3]	SOS 3, numerator: $a_{0,3}$ $a_{1,3}$, $a_{2,3}$
+71	T_SINT16	&a_iir4x_sos_filter[3]->a_iir4x_sos_num_form	SOS 3, Q-format of $a_{0,3}$ $a_{1,3}$, $a_{2,3}$
+72	T_SINT16	&a_iir4x_sos_filter[4]->a_iir4x_sos_fact	SOS 4, scaling factors G_4'

&	Type	Variable Name	Description
+73	T_SINT16	&a_iir4x_sos_filter[4]->a_iir4x_sos_fact_form	SOS 4, Q-formats of G_1'
+74	T_SINT16	&a_iir4x_sos_filter[4]->a_iir4x_sos_den[2]	SOS 4 denominator: $b_{1,4}$, $b_{2,4}$
+76	T_SINT16	&a_iir4x_sos_filter[4]->a_iir4x_sos_num[3]	SOS 4, numerator: $a_{0,4}$ $a_{1,4}$, $a_{2,4}$
+79	T_SINT16	&a_iir4x_sos_filter[4]->a_iir4x_sos_num_form	SOS 4, Q-format of $a_{0,4}$ $a_{1,4}$, $a_{2,4}$
+80	T_SINT16	&a_iir4x_sos_filter[5]->a_iir4x_sos_fact	SOS 5, scaling factors G_5'
+81	T_SINT16	&a_iir4x_sos_filter[5]->a_iir4x_sos_fact_form	SOS 5, Q-formats of G_5'
+82	T_SINT16	&a_iir4x_sos_filter[5]->a_iir4x_sos_den[2]	SOS 5, denominator: $b_{1,5}$, $b_{2,5}$
+84	T_SINT16	&a_iir4x_sos_filter[5]->a_iir4x_sos_num[3]	SOS 5, numerator: $a_{0,5}$ $a_{1,5}$, $a_{2,5}$
+87	T_SINT16	&a_iir4x_sos_filter[5]->a_iir4x_sos_num_form	SOS 5, Q-format of $a_{0,5}$ $a_{1,5}$, $a_{2,5}$
+88	T_SINT16	&a_iir4x_sos_filter[6]->a_iir4x_sos_fact	SOS 6, scaling factors G_6'
+89	T_SINT16	&a_iir4x_sos_filter[6]->a_iir4x_sos_fact_form	SOS 6, Q-formats of G_6'
+90	T_SINT16	&a_iir4x_sos_filter[6]->a_iir4x_sos_den[2]	SOS 6, denominator: $b_{1,6}$, $b_{2,6}$
+92	T_SINT16	&a_iir4x_sos_filter[6]->a_iir4x_sos_num[3]	SOS 6, numerator: $a_{0,6}$ $a_{1,6}$, $a_{2,6}$
+95	T_SINT16	&a_iir4x_sos_filter[6]->a_iir4x_sos_num_form	SOS 6, Q-format of $a_{0,6}$ $a_{1,6}$, $a_{2,6}$

Table 2.7 The IIR 4.x API Parameters – SOS Part

&	Type	Variable Name	Format	Range	Comments
+00	T_UINT16	d_iir4x_control	Q0	0x0000	Disable
				0x0001	Enable 8kHz
				0x0002	Enable 16kHz
+01	T_UINT16	d_iir4x_frame_size	Q0	0x0050	80 samples (10ms@ 8kHz)
				0x00A0	160 samples (default value) (2x10ms @ 8kHz, 10ms @16kHz)
				0x0140	320 samples (2x10ms @16kHz)
+02	T_UINT16	d_iir4x_fir_swap	Q0	0x0001	FIR acting before IIR (SOS)
				0x0002	IIR SOS acting before FIR
+03	T_UINT16	d_iir4x_fir_enable	Q0	0x0000	FIR OFF
				0x0001	FIR ON
+04	T_UINT16	d_iir4x_fir_length	Q0	[0x0001… 0x0028]	Min = 1 tap
					Max = 40 taps, $N = 40$.
+05	T_SINT16	d_iir4x_fir_shift	Q0	[0x0000… 0x7FFF]	$Q(\boldsymbol{H}_N)$. See note (1).
+06	T_SINT16	a_iir4x_fir_taps[40]	$Q(\boldsymbol{H}_N)$	[0x8000… 0x7FFF]	$\{h_i\}, 0 \le i \le N-1$. See note (1).
+46	T_UINT16	d_iir4x_sos_enable	Q0	0x0000	IIR (SOS) disable
				0x0001	IIR (SOS) enable
+47	T_UINT16	d_iir4x_sos_number	Q0	[0x0001…0 x0006]	Min = 1 SOS (biquad)
					Max = 6 SOS (biquads), $K = 6$.
+48	T_SINT16	a_iir4x_sos_fact	$Q(\boldsymbol{G}_k^{'})$	[0x8000… 0x7FFF]	SOS 1, $k = 1$. See note (2).
+49	T_SINT16	a_iir4x_sos_fact_form	Q0	[0x0000… 0x7FFF]	$Q(\boldsymbol{G}_k^{'})$, $k = 1$. See note (2).
+50	T_SINT16	a_iir4x_sos_den[2]	Q14	[0x8000… 0x7FFF]	SOS 1 denominator: $b_{1,1}$, $b_{2,1}$
+52	T_SINT16	a_iir4x_sos_num[3]	$Q(\boldsymbol{a}_k)$	[0x8000… 0x7FFF]	SOS 1, numerator: $a_{0,1}$ $a_{1,1}$, $a_{2,1}$ See note (3).
+55	T_SINT16	a_iir4x_sos_num_form	Q0	0x0000… 0x7FFF]	$Q(\boldsymbol{a}_k)$, $k = 1$. See note (3).
…	…	…	…	…	…
+88	T_SINT16	a_iir4x_sos_fact	$Q(\boldsymbol{G}_k^{'})$	[0x8000… 0x7FFF]	SOS 6, $k = 6$. See note (2).
+89	T_SINT16	a_iir4x_sos_fact_form	Q0	[0x0000…	$Q(\boldsymbol{G}_k^{'})$, $k = 6$. See note (2).

&	Type	Variable Name	Format	Range	Comments
				0x7FFF]	
+90	T_SINT16	a_iir4x_sos_den[2]	Q14	[0x8000… 0x7FFF]	SOS 6, denominator: $b_{1,6}$, $b_{2,6}$
+92	T_SINT16	a_iir4x_sos_num[3]	$Q(\boldsymbol{a}_k)$	[0x8000… 0x7FFF]	SOS 6, numerator: $a_{0,6}$ $a_{1,6}$, $a_{2,6}$ See note (3).
+95	T_SINT16	a_iir4x_sos_num_form	Q0	0x0000… 0x7FFF]	$Q(\boldsymbol{a}_k)$, $k=6$. See note (3).
+96	T_SINT16	d_iir4x_gain	Q13	0x0000	Digital gain, muting mode (-inf dB)
				[0x0001… 0x2000… 0x7FFF]	[-78dB (min value)… 0dB (default value)… +12dB (max value)]. See note (4).

Table 2.8 The IIR 4.x API Parameters Format and Range

(1) Assuming that FIR impulse response is length N :

$$\boldsymbol{H}_N = \left[h_0, \cdots h_{N-1}\right]^{\mathrm{T}}, \tag{2.1}$$

the FIR taps are 16-bits signed quantized with Q-format corresponding to:

$$Q\left(\boldsymbol{H}_N\right) = -\underset{\mathbb{Z}^-}{Inf}\left[\log_2\left(\max\left\{\left|\boldsymbol{H}_N\right|\right\}\right)\right], \tag{2.2}$$

leading to:

$$\text{a\_iir4x\_fir\_taps}[i] = \text{round}\left\{h_i.2^{15+Q(\boldsymbol{H}_N)}\right\}, 0 \le i \le N-1. \tag{2.3}$$

Example:

Assuming one FIR designed with 11 taps in floating point representation:

$\boldsymbol{H}_N$ = [0.0004, 0.0021, 0.0139, 0.0465, 0.0992, 0.1410, 0.0992, 0.0465, 0.0139, 0.0021, 0.0004]$^{\mathrm{T}}$,

we have:

$$\max\left\{\left|\boldsymbol{H}_N\right|\right\} = 0.1410,\ \log_2\left(\max\left\{\left|\boldsymbol{H}_N\right|\right\}\right) = \text{-}1.9589 \pm 1.10^{-4},\ \underset{\mathbb{Z}^-}{Inf}\left[\log_2\left(\max\left\{\left|\boldsymbol{H}_N\right|\right\}\right)\right] = -2,\ Q\left(\boldsymbol{H}_N\right) = 2$$

and so quantization of FIR is done with $\text{a\_iir4x\_fir\_taps}[i] = \text{round}\left\{h_i.2^{17}\right\}, 0 \le i \le N-1$, leading to:

a_iir4x_fir_taps =
{0x0034, 0x0113, 0x071D, 0x17CE, 0x32CA, 0x4831, 0x32CA, 0x17CE, 0x071D, 0x0113, 0x0034}.

(2) Assuming that $G_k^{'}$ denotes the scaling factor of the k^{th} Second Order Section (SOS), the quan tization of $G_k^{'}$ is done using $Q(\boldsymbol{G}_k^{'})$ Q-format as described bellow:

$$\text{a\_iir4x\_sos\_fact}[k] = \text{round}\left\{G_k^{'}.2^{15-Q(G_k^{'})}\right\}, 1 \le k \le K. \tag{2.4}$$

(3) Assuming that $\boldsymbol{a}_k$ denotes the numerator of the k^{th} Second Order Section (SOS), the quantiza tion of $\boldsymbol{a}_k$ is done using $Q(\boldsymbol{a}_k)$ Q-format as described below:

$$\text{a\_iir4x\_sos\_num}[i+3.(k-1)] = \text{round}\left\{a_{i,k}.2^{15-Q(\boldsymbol{a}_k)}\right\}, 0 \le \text{i} \le 2, 1 \le k \le K. \quad (2.5)$$

(4) Digital gain is applied at the output of equalization filter. From D_{dB} specification, the quantization is done as follow:

$$\text{d\_iir4x\_gain} = 2^{13}.10^{\frac{D_{dB}}{20}}. \quad (2.6)$$

Appendices

A. Appendix: IIR 4.x Module Interface & Block Diagram

The IIR 4.x interface (API) is presented below (Table 2.9) in connection with the functional block diagram of the module (Figure 2.1).

&	Type	Variable Name	Description
+00	T_UINT16	d_iir4x_control	Equalization enable/disable
+01	T_UINT16	d_iir4x_frame_size	Equalization IO frame size
+02	T_UINT16	d_iir4x_fir_swap	FIR/IIR structures swap
+03	T_IIR4X_FIR_PARAM	d_iir4x_fir_filter	FIR parameters
+46	T_IIR4X_SOS_PARAM	d_iir4x_sos_filter	IIR (SOS) parameters
+96	T_SINT16	d_iir4x_gain	Digital gain after equalization (D)

Table 2.9 The IIR 4.x API Parameters Description (Recall)

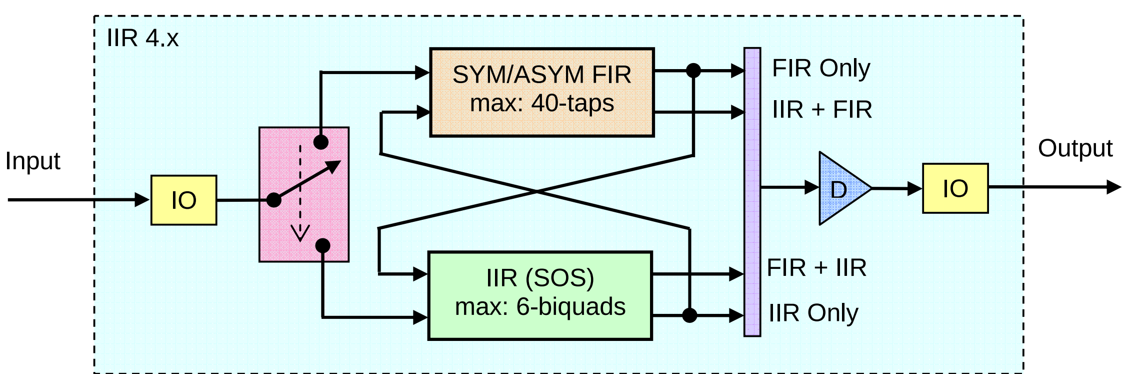

Figure 2.1 The IIR 4.x Block Diagram

B. Appendix: API Parameters & Numerical Filter Structures

The relation between some API parameters in IIR 4.x is presented below for FIR structure (Figure 2.2) and IIR (SOS) structure (Figure 2.3).

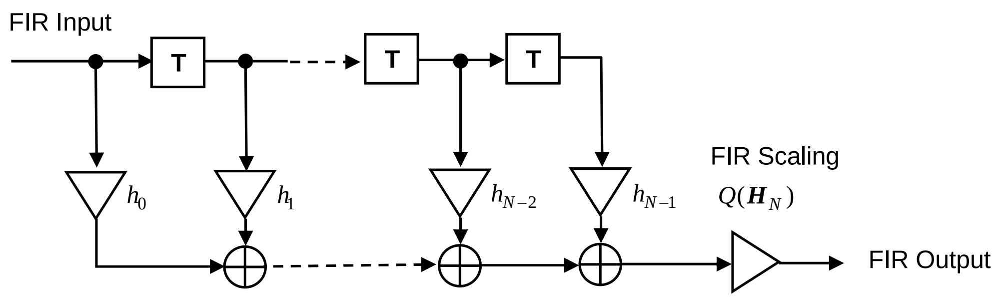

Figure 2.2 The IIR 4.x – Implemented FIR Structure

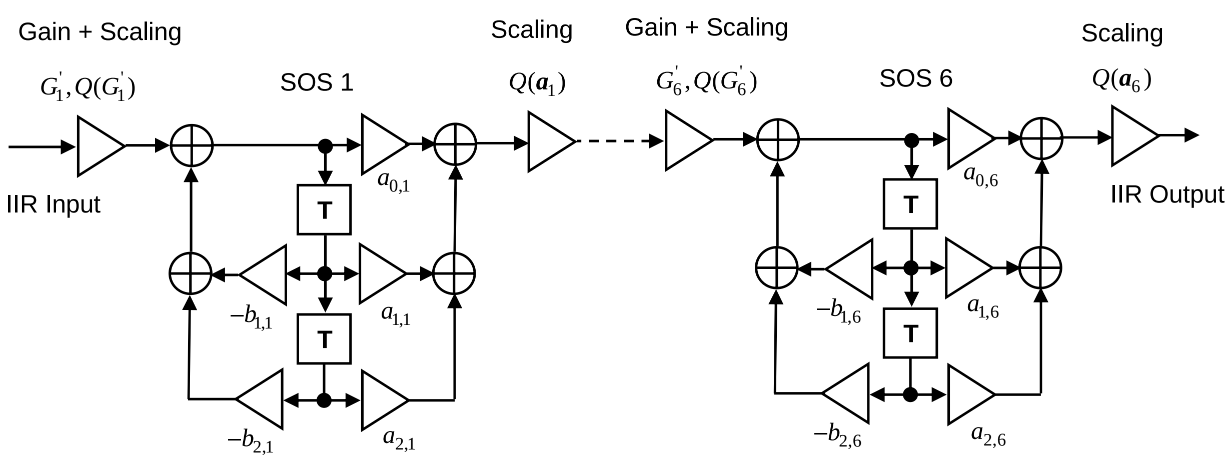

Figure 2.3 The IIR 4.x – Implemented IIR Structure (SOS)