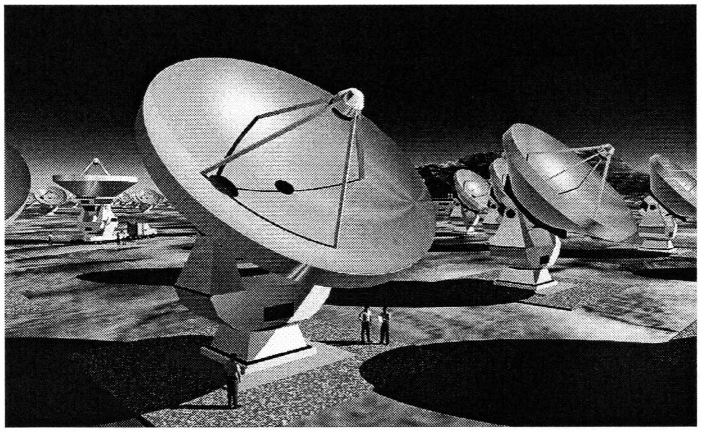
Author:	A.W. Gunst	Date of issue:	March 21, 2002	Scope:	Development
Verified by:	J.D. Bregman	Kind of issue:	Public	Doc.nr.:	Report-004 ASTRON-28000-R1
Responsible:	C.M. de Vos	Status:	Final	File:	G:\data\tekkamer\28000\
Approved by:	C.M. de Vos	Revision nr.:	2.1		53Mrepor.pdf
The ALMA FC study carried out at ASTRON is a contribution to the European ALMA Backend Electronics Team.					

FIR Filter Size in ALMA



Document history:

Revision	Date	Section	Page	Modification
0.1	April 24, 2001	-		Creation
0.2	April 26, 2001	-		Continued
0.3	June 1, 2001	-		First correction (incl. title)
1.0	June 11, 2001	-		Released
1.1	July 2, 2001	-		Second correction
2.0	February 22, 2002	-		A note of the author added
2.1	March 21, 2002	-		Conclusion revised

Note of the Author

In Section 5 of release 1 of this document the real estate of the ALMA FIR filters was estimated using the Altera tool Quartus II [1] in combination with the FIR compiler 2.0. I covered myself by a footnote which reveals that the first order approximation of the FIR compiler does **not** always represent the real implementation. During several mail conversations with the Altera helpdesk, it was concluded that the FIR compiler 2.0 [2] does not generate the correct estimation as a function of truncated LSB and MSB. The bug was fixed in the new FIR compiler, version 2.2.2. In this compiler, there was no gain anymore by truncating the filter result. Therefore the results as function of the number of bits in Section 5 are obsolete. However, implementing the filters of ALMA with the FIR compiler is not efficient. We are better of to implement the low bit filter with Look Up Tables as proposed in [3]. Then, the number of RAM bits to be used equals

$$2^{n_i} \cdot n_e \cdot N_t, \tag{1}$$

with n_i the number of input bits, n_e the number of output bits and N_t the number of taps. The result of all possible input combinations multiplied with each coefficient is stored in RAM.

Contents

Introduction 5

1 FIR Filter Design 5

2 Estimation of Low Pass FIR Filter Order 7

3 Filter Types 9

4 Ripple and Attenuation Influences 10

5 ALMA FIR Filter Real Estate 13

Conclusion 17

References 18

Introduction

A number of backend problems have to be addressed during the ALMA study [4]. One of these problems is the taps-lags relation. The study of the relation between FIR filter and correlator must prevent overspecifying one of these systems compared with the other. Before doing this, some feeling of correlator size and FIR filter size given a set of specifications is necessary. This report deals with FIR filter size. The goal of this report is to give readers a feeling of the FIR filter size when specifications are varied. The order of the FIR filter is associated with the size. To enhance the awareness of the FIR filter size, some Logic Elements counts are given in case the filter was really implemented on an FPGA. This document is focused on the size of one FIR filter and not on a bank of filters. Therefore FIR filter bank architecture discussions are not dealt with in this document. Nevertheless this is an important issue for the total FIR filter size and should be a topic of further study. Only for a straightforward FIR filter bank implementation the total Logic Element count is given.

The first section starts with defining the relevant quantities for FIR filters. These quantities are used throughout the report. Also the used algorithm for designing digital FIR filters is discussed here. Section 2 deals with two estimators, which can be used to estimate the order of the filter given a specification. In Section 3 the size is determined for three types of real linear phase FIR filters. The size is discussed as a function of transistion region. Section 4 deals with low pass filters and discusses the size as function of passband ripple and stopband attenuation. Finally Section 5 discusses the hardware requirements for the current ALMA specifications and overviews it as a function of input bits, coefficient bits and output bits.

1 FIR Filter Design

In general five quantities are important in filter design. Fixing four quantities and choosing a specific algorithm to design the filter, results in the fifth quantity. In this report the following four quantities are tuned:

- Passband frequency F_p in fraction of f_s
- Stopband frequency F_s in fraction of f_s
- Passband ripple R_p in dB and δ_1 (defined later up)
- Stopband attenuation R_s in dB and δ_2 (defined later up)

This results in the fifth quantity: the order N. The context or an explicit reference makes clear which ripple or attenuation is meant in the text. The definition of the quantities is given in Fig. 1 (sample frequency is denoted as f_s). A related quantity is the transistion region, which is defined as follows:

$$t_r = (F_s - F_p) \cdot f_s \tag{2}$$

The tuning of passband and stopband frequency is done indirectly via the transistion region (defined in Eq. (2)).

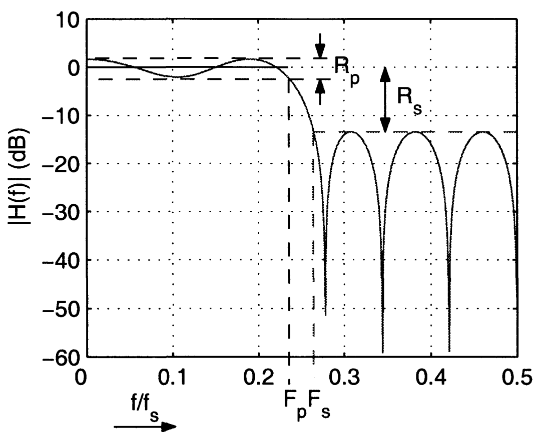


Figure 1: Definition of FIR filter quantities, with f_s the sample frequency.

For desiging filters the SPtool of Matlab is used [5]. With this tool several methods to design filters can be chosen. In this document results with the equiripple FIR filter design are obtained. With an equiripple filter design the error is uniformly distributed in the passband and in the stopband. It turns out that given the four input arguments the order of an equiripple filter has the smallest possible order N [6]. The design of equiripple filters is done via the Remez exchange algorithm, also known as the McClellan-Parks algorithm. As input arguments the four quantities mentioned before are used.

2 Estimation of Low Pass FIR Filter Order

In literature also formulas are available, which estimates the order of a linear phase equiripple lowpass filter. The most famous ones are the Kaiser's formula [6]:

$$N = \frac{-20\log\sqrt{\delta_1\delta_2} - 13}{14.6\text{tr}} \cdot f_s \approx \frac{-10.27\log(\delta_1\delta_2) - 13.36}{15\text{tr}} \cdot f_s \tag{3}$$

and Bellanger's formula [6]:

$$N = \frac{-2\log(10 \cdot \delta_1\delta_2)}{3\text{tr}} \cdot f_s \approx \frac{-10\log(\delta_1\delta_2) - 10}{15\text{tr}} \cdot f_s, \tag{4}$$

where

$$\delta_1 = \frac{10^{\frac{R_p}{20}} - 1}{2} \tag{5}$$

$$\delta_2 = 10^{-\frac{R_s}{20}} \tag{6}$$

Both δ's are depicted in Fig. 2 (adopted from [6]).

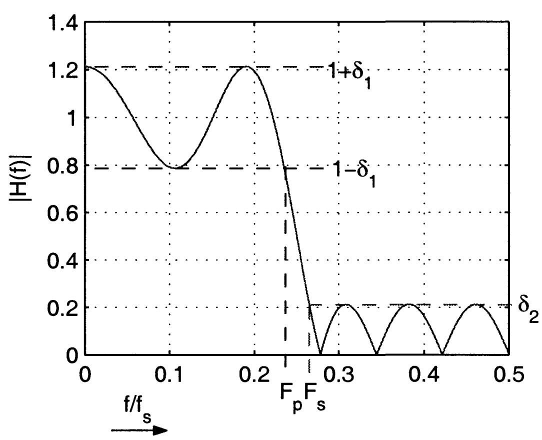


Figure 2: Definition of additional FIR filter quantities, with f_s the sample frequency.

The order N has to be an integer number, therefore the results of both estimations are roundend up to the nearest integer. From both equations can be observed that a ripple δ_1=0.01 (R_p=0.17 dB) has the same impact on the number of taps than a stopband attenuation δ_2=0.01 (R_s=40 dB). So, the penalty of going from 1 dB ripple to 0.17 dB ripple is relative low (40 dB attenuation in the stopband assumed). Assuming a transistion region of $0.025f_s$ results for a ripple of 0.17 dB in 53 and 60 taps using respectively Eq. 3 and Eq. 4, while a ripple of 1 dB gives 74 and 80 taps. The smallest δ is the strongest component in the number of taps. Worst case the other δ can increase the number of taps (constant term is omitted) with a factor of approximately two (than $\delta_1 = \delta_2$ applies).

3 Filter Types

In SPtool four types of filters can be chosen: lowpass, highpass, bandpass and bandstop filters. The size of the first three as function of transistion region are determined in this section. For this a fixed ripple R_p of 0.1 dB and a suppression R_s of 40 dB is assumed. Further the transistion region of the lowpass and highpass filters are chosen symmetric around $0.25f_s$, while the total transistion region for the bandpass filters is split in two regions of 0.5tr symmetric around $0.125f_s$ and $0.375f_s$.

The results after simulation and using both formula's for estimating the order are shown in Table 1. In Fig. 3(a) the order of the three types of filters is plotted as a function of transistion region (expressed in f_s). As expected the difference between the lowpass design and highpass design is not significant. For the bandpass filter roughly two times the amount of taps is necessary. This is because two transistion regions are required for realizing the bandpass. You can also put it in another way: for the same size the transistion region is two times as large.

The approximation of the Kaiser and Bellanger formula and the simulation for a low pass filter is depicted in Fig. 3(b). From this can be seen that the formula's are a good approximation, especially for small transistion regions. The Kaiser formula gives a lower bound, while the Bellanger formula looks to give an upper bound for small transition regions. Both figures show that the order is approximately linear with transistion region. Doubling the transistion region gives a size increment of two. This is more valid for small transistion regions. The linearity can also be observed by inspecting Eq. (3) and Eq. (4), where tr is in the denumerator.

tr	N (lowpass)	N (highpass)	N (bandpass)	N (Kaiser)	N (Bellanger)
$0.32f_s$	4	4	12	7	7
$0.16f_s$	12	12	26	13	14
$0.08f_s$	26	26	53	26	27
$0.04f_s$	53	52	106	51	54
$0.02f_s$	106	106	211	101	108
$0.01f_s$	211	210	422	202	216
$0.005f_s$	422	418	843	403	432
$0.0025f_s$	843	836	1686	805	864

Table 1: FIR filter order as a function of transistion region for lowpass, highpass and bandpass filter. Also the Kaiser and Bellanger estimations are given. For the lowpass and highpass filter the transistion region is around $0.25f_s$, while the total transistion region for the bandpass filters is split in two regions around $0.125f_s$ and $0.375f_s$ (R_p=0.1 dB and R_s=40 dB).

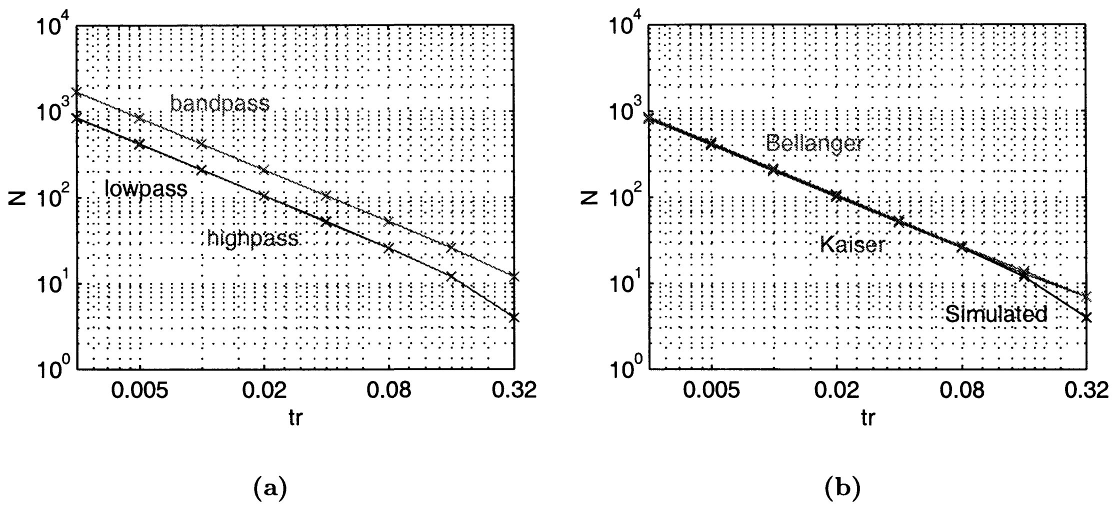


Figure 3: *Logarithmic plot of order as a function of transistion region (expressed in f_s) for (a) lowpass, highpass and bandpass filter and (b) for lowpass filter obtained via simulation, Kaiser's approximation and Bellanger's approximation (R_p=0.1 dB and R_s=40 dB).*

4 Ripple and Attenuation Influences

In this section the transistion region is set at $0.05f_s$. Only the results for lowpass filters is discussed. The effect of changing the ripple specification can best be seen in terms of δ_1 instead of R_p. Results are listed in Table 2 and depicted in Fig. 4 for a suppression R_s of 40 dB.

δ_1	R_p(dB)	N (lowpass)	N (Kaiser)	N (Bellanger)
0.0025	0.04	47	46	49
0.005	0.09	43	42	45
0.01	0.17	39	37	40
0.02	0.34	35	33	36
0.04	0.67	31	29	32
0.08	1.29	27	25	28
0.16	2.41	23	21	24
0.32	4.30	19	17	20

Table 2: FIR filter order as a function of the ripple δ_1 and R_p (suppression R_s is 40 dB) for a lowpass filter with a transistion region of $0.05f_s$ symmetric around $0.25f_s$. The results are given for a simulation, and according to the Kaiser and Bellanger estimations.

Fig. 4 illustrates the effect of ripple specification on the order. From this the lower and upper bound of both approximations is seen again. The simulation shows a constant decrease of 4 in size when the ripple δ_1 is doubled. When the transistion region is doubled than the decrease in size is 2 when doubling the ripple δ_1. So, the cost of reducing the ripple δ_1 is therefore dependent of the transistion region. This can be seen from Eq. 3 and 4 as well. The increase of the number of orders when the ripple is doubled results for Kaiser:

$$\frac{-10.27 \log 2}{15\mathrm{t}} \tag{7}$$

and for Bellanger:

$$\frac{-10 \log 2}{15\mathrm{t}} \tag{8}$$

Both yields an increase of 4. This depends inverse proportional on tr.

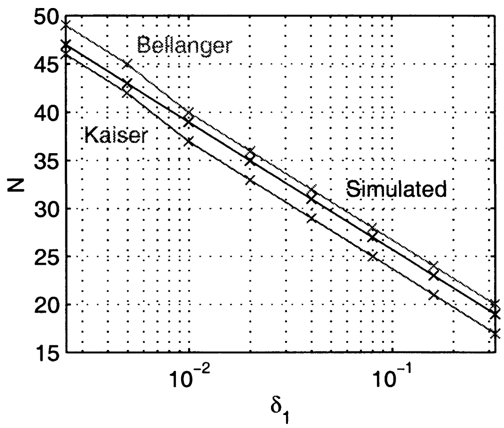


Figure 4: Order N as a function of ripple δ_1 (R_s=40 dB and tr=0.05f_s).

From now on the ripple R_p is fixed to 0.1 dB again and the suppression is varied. For this δ_2 is changed. Reducing δ_2 with 2 results in an attenuation reduction of 6 dB. From Eq. (3) and (4) can be seen that the effect must be the same as for the ripple δ_1. After simulation and calculating the order Table 3 was obtained. The results are depicted in Fig. 5. Again an almost linear line is obtained. Increasing the attenuation with 6 dB results in an order increment of approximately 4 (for the Bellanger formula this is exactly the case). A change in transistion region has the same effect as already mentioned for the ripple δ_1. Doubling the transistion region, results in an order increase of 2 when the attenuation R_s must be 6 dB higher.

δ_2	R_s(dB)	N (lowpass)	N (Kaiser)	N (Bellanger)
0.0025	52	49	49	52
0.005	46	46	45	48
0.01	40	42	41	44
0.02	34	38	37	40
0.04	28	35	32	36
0.08	22	31	28	32
0.16	16	27	24	28
0.32	10	24	20	24

Table 3: FIR filter order as a function of the attenuation δ_2 and R_s (ripple R_p=0.1 dB) for a lowpass filter with a transistion region of $0.05f_s$ symmetric around $0.25f_s$. The results are given for a simulation, and according to the Kaiser and Bellanger estimations.

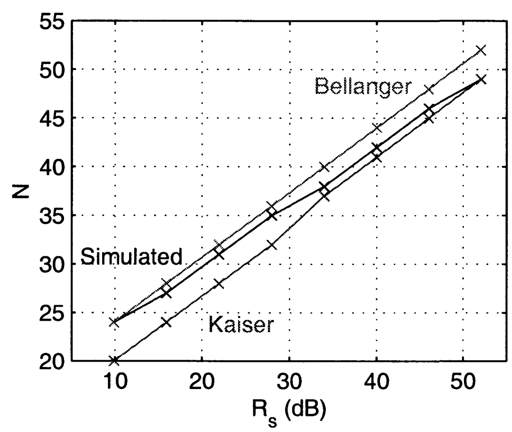


Figure 5: Order N as a function of attenuation R_s in dB (R_p=0.1 dB and tr=$0.05f_s$).

In Fig. 6 the order as function of the ripple and attenuation are given. From this figure can be seen that ripple and attenuation is exchangable as predicted by Eq. (3) and (4). This does not hold for very small ripples (the graph is not flat anymore) . In this region some extra taps must be added for gaining a performance improvement. Given the transistion region a ripple δ_1 improvement of two times cost the same as an attenuation δ_2 improvement of two times. For ALMA the ripple can be calibrated out, and in that sense we must not save on ripple rather than attenuation.

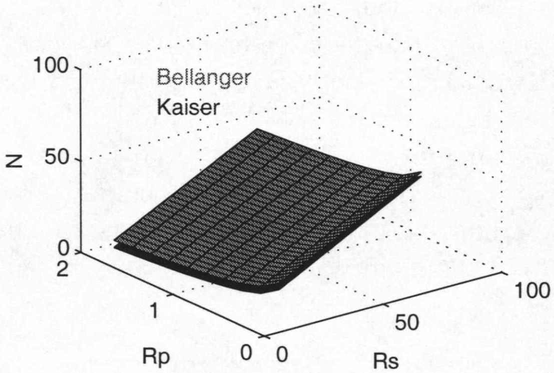


Figure 6: Order N as a function of ripple R_p and attenuation R_s, both in dB, for a transistion region of $0.05f_s$.

5 ALMA FIR Filter Real Estate

The previous sections treated the FIR filter size in a general way. This section tends to be more specific for the ALMA system. To express the loss of the transistion region it can be written as a percentage of the total passband:

$$\mathrm{tr}(\%) = \frac{\mathrm{tr}}{\mathrm{passband}} \cdot 100\% \tag{9}$$

Suppose 2 subbands are required in ALMA (passband $0.25f_s$) and the transistion region equals 10 percent in total, i.e. tr= $0.025f_s$. The attenuation must be larger than 40 dB and the ripple smaller than 1 dB. Filling in these number in the upper bound, results for Bellanger's formula (4) in 61 taps (order + 1).

In ALMA a 2 GHz frequency band is divided in 32 subbands [7]. To be more general, each passband is $\frac{0.5f_s}{32} = \frac{f_s}{64}$, with f_s the sample frequency. Compared with the previous example the number of subbands is increased with a factor 16, while the 10 percent transistion region still has to hold. For bandpass filters two transistion regions of 10 percent each is assumed. Using the fact that the number of taps grows linearly with the absolute transistion regions results in the conclusion that the number of taps has to be 16 times 61 (Bellanger's formula gives an order of 945 because of the rounding up operation). From a simulation using a lowpass filter an order of 908 was obtained. The transfer characteristic is depicted in Fig. 7. Putting the filter coefficients (8 bits) in the FIR compiler [2] of Quartus II[1] [1] and assuming **three** input bits and full precision

[1]Quartus II was used to get a quick impression of the real estate of FIR filters in ALMA.

at the output (17 bit) results in 11440 Logic Elements[2] (parallel implementation). Therefore the complete filterbank occupies $32 \cdot 11440 = 366$k Logic Elements. The largest APEX20KE device (EP20K1500E) has a maximum of 51840 Logic Elements (corresponds with approximately 1.5 million gates). So, at least 7 FPGA's will be necessary to implement the total filterbank in this way. The output frequency of the FIR filters in ALMA is 125 MHz, while the input frequency is 4 GHz. Therefore the multipliers can run on 125 MHz as well [6].Furthermore, it is possible to exploit the architecture. This must be a topic of a follow up document about FIR filters in ALMA. Implementing the filters on ASIC's in stead of FPGA's will reduce the area necessary. Especially the power consumption will go down signifcantly.

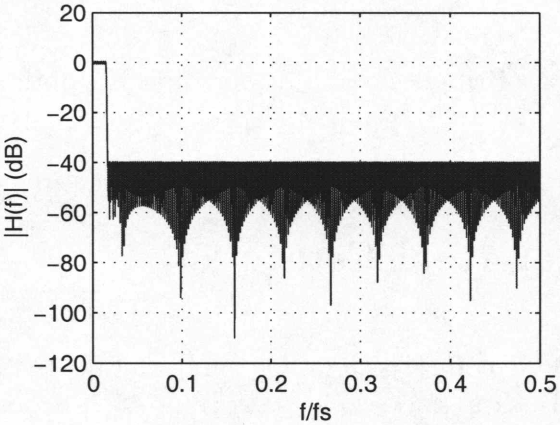


Figure 7: Frequency characteristic of small low pass filter with 10 percent transistion region, 40 dB suppression and 1 dB ripple.

The size will also go down when the amount of output bits is reduced. Dependent on the input signal MSB's and LSB's can be skipped. For the following results it is assumed that the output is truncated when bits are removed.

A number of situations is put into the FIR compiler to gain a feeling of the size of FIR filters for ALMA when changing the number of bits. The same specification as mentioned at the start of this section is used as reference. The effect of removing MSB's and LSB's is shown in Table 4. The results are graphically shown in Fig. 8(a). From this can be concluded that the size reduction is the same for skipping two MSB's or one LSB. So skipping a LSB has double the amount of impact compared with skipping a MSB. Reducing the amount of output bits with two, results in a hardware reduction larger than two (for the region where more than 5 bits are already removed).

[2] The number of Logic Elements are obtained from the FIR compiler, rather than a real implementation. Therefore the number is a first order approximation and will **not** always represent the real implementation.

# bit removed	LE (LSB)	LE (MSB)
0	11440	11440
1	11230	11340
2	11000	11230
3	10700	11120
4	10270	11000
5	9700	10870
6	8920	10700
7	8080	10500
8	7240	10270
9	6410	10010
10	5570	9700
11	4730	9330
12	3890	8920
13	3050	8500
14	2220	8080
15	1380	7660
16	540	7240

Table 4: The number of Logic Elements (LE), when bits are removed from the LSB side and the MSB side. The number of output bits for full output precision equals 17.

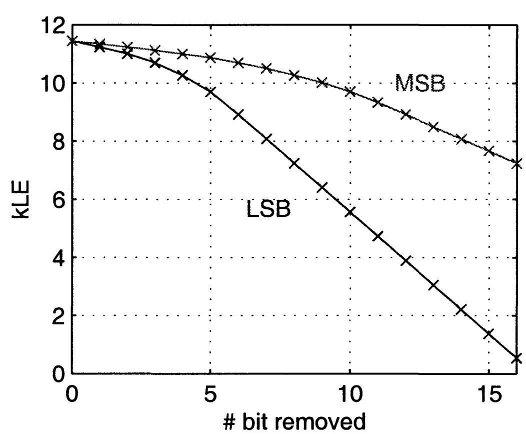


Figure 8: Number of Logic Elements (LE) as function of removed bits for Most Significant Bits and Least Significant Bits (kLE=1000·LE).

The calculation of the number of FPGA's required for the ALMA filterbanks is repeated now for having 3 output bits (14 LSB's are removed) instead of full precision. The number of Logic

Elements for one FIR filter equals 2220. Multiplying this with 32 (the number of filters) and with 16 (because of the clock reduction) results in 1137k Logic Elements, which fit into 22 FPGA's. The structure of the filterbank is also for this case not exploited.

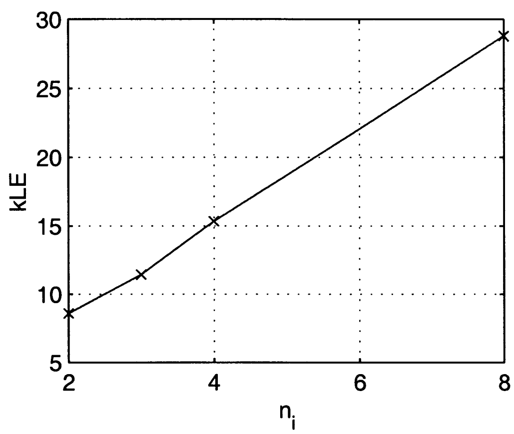
Table 5 shows the dependency as function of number of input bits with 8 coefficient bits, while Table 6 does the same for the coefficient bits with three input bits. The results are also depicted in Fig. 9. Doubling the number of input bits results in approximately a double size. This rule is not valid for the number of coefficient bits. The reason for this could be the mapping of the functionality on the hardware. For example the symmetrical architecture selection [2]: The FIR compiler examines the coefficients and determines the filters symmetry. From this an optimum algorithm is used which minimizes the amount of computation. For symmetric filters two samples are added prior to multiplication, saving one multiplication operation.

n_i	LE
2	8600
3	11440
4	15350
8	28800

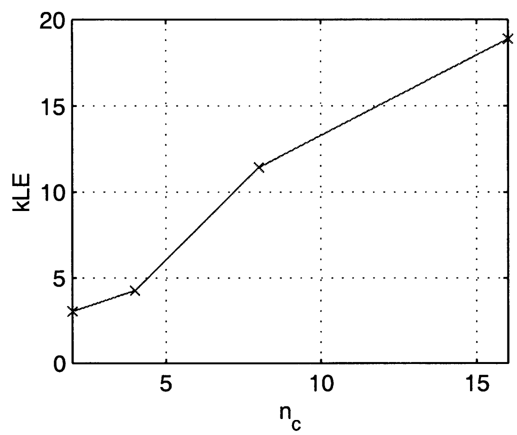
Table 5: The number of Logic Elements (LE) as function of the number of input bits (n_i), with the number of coefficient bits n_c=8.

n_c	LE
2	3030
4	4260
8	11440
16	18880

Table 6: The number of Logic Elements (LE) as function of the number of coefficient bits (n_c), with n_i=3.



(a)



(b)

Figure 9: *Logic Elements as function of (a) input bits n_i and (b) coefficient bits n_c (kLE=1000·LE).*

The impact of reducing the amount of bits on performance at several stages is discussed in [8]. For lower bit FIR filters a seperate document will be produced [9]. The number of coefficient bits must be determined dependent on which distortion is allowed in the transfer characteristic. In general this will be related with the required attenuation in the stopband.

Conclusions

It can be concluded that the size of equiripple linear FIR filters is inverse proportional on the size of the transistion region. Halving the transistion region, results in a size increase of two. The effect of varying the ripple and attenuation on size appeared dependent on transistion region as well. For a transistion region of $0.05f_s$ a size reduction of 4 was found when the ripple δ_1 is doubled. If the transistion region was halved the size decrease when doubling the ripple δ_1 was 8.

Two common used estimators of the filter order for equiripple lowpass filters are Kaiser's estimation and Bellanger's estimation. From the simulations performed for this report, it appeared that the Bellanger's formula was an upper bound, while Kaiser's formula was an lower bound. In most cases the simulation results lay in between. This rule was violated for transistion regions larger than $0.08f_s$, with f_s the sample frequency.

The total hardware occupied by the FIR filters depend on the used architecture. When a parallel implementation for the ALMA FIR filterbank is used with 3 input bits, 8 bit coefficients and full precision output (17 bit) 7 large FPGA's are required [3]. Because the output frequency of the FIR filters in ALMA is 125 MHz, the multipliers can run on 125 MHz as well.

The combination of knowing which hardware is required for a set of specifications and knowing the performance gained is vital in choosing the right specification. We must prevent over specifying the FIR filter specifications when the number of coefficient bits for example is very low. Demanding a very large attenuation, while truncating the coefficients to two bit would be a waste of logic. For the total size the architecture of the filterbank plays also an important role. The FIR filter bank architecture must therefore be next topic of study.

References

[1] *Quartus II.* ALTERA, 2001. Reference: `www.altera.com`.

[2] *FIR Compiler MegaCore Function.* ALTERA, february 2001. Reference: `www.altera.com`.

[3] B. Quertier. Architecture and tests of FPGA_1 prototype. *ALMA Report*, June 2001.

[4] Gunst & Bregman. Backend design problems. Technical Report Report-003 ASTRON-28000-R1, ASTRON, Dwingeloo, the Netherlands, April 2001.

[5] *Matlab: High Performance Numeric Computation and Visulation Software.* The Math Works Inc., 2000. Reference: `www.mathworks.com`.

[6] P.P. Vaidyanathan. *Multirate Systems and Filter Banks.* Prentice Hall, Englewood Cliffs, 1993.

[7] A.W. Gunst. FIR specification meeting. Technical Report ASTRON-ALMAFC-00256, ASTRON, Dwingeloo, the Netherlands, April 2001.

[8] A.W. Gunst. Simulation results of finite length FIR filters. Technical Report Report-002 ASTRON-28000-R1, ASTRON, Dwingeloo, the Netherlands, November 2000.

[9] A.W. Gunst. Simulation results of low bit FIR filters. Technical Report Report-005 ASTRON-28000-R1, ASTRON, Dwingeloo, the Netherlands, April 2001.

[3] This is a first order approximation and based on estimations of Altera's FIR compiler which does **not** always represent the reality.