Design Automation for Embedded Systems, 9, 123–139, 2004.

Moduli Set Selection and Cost Estimation for RNS-Based FIR Filter and Filter Bank Design

YONG LIU
EDMUND M.-K. LAI* asmklai@ntu.edu.sg
School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798

Abstract. Moduli selection is one of the most important issues in the implementation of systems that make use of residue number systems. In this paper, we describe a software tool that assists system designers in moduli selection for the design of RNS-based FIR filters and filter banks. According to some filter specification parameters, the software tool constructs valid moduli sets and calculates their estimated implementations cost in terms of delay, area and power consumption based on results obtained in logic synthesis. Moduli set that is most suitable for the user requirements is selected, together with the estimated cost, to be the output. Outputs of the software tool also indicate that certain level of trade-off among delay, area and power consumption exists for the RNS-based filter and filter bank implementation by using different moduli sets.

Keywords: residue number system, FIR filter, moduli set selection

1. Introduction

Residue number systems (RNS) are considered suitable for the implementation of high-speed digital signal processing devices due to their inherited parallelism, modularity, fault tolerance and localized carry propagation properties. Some arithmetic operations, such as addition and multiplication, can be carried out more efficiently in RNS than in conventional two's complement systems. Thus, residue number systems are particularly suitable to implement finite impulse response (FIR) filters, and hence FIR filter banks, due to the fact that FIR filtering essentially only require multiplications and additions.

The choice of moduli set is one of the most important considerations when designing RNS-based FIR filter banks. Given a desired dynamic range, there are two main approaches for choosing a moduli set [5]. The first approach is to use a small number of moduli of the form 2^n and $2^n \pm 1$. The other approach is to use a larger set of moduli that is primarily made up of small prime numbers. In [14], a scheme for moduli set selection is proposed, which selects moduli set that results in the most efficient residue-to-binary converter structure based on the required dynamic range. However, the choice of moduli set not only affects the hardware complexity and delay of the residue-to-binary converter, but also has an impact on the hardware complexity and delay of other components such as binary-to-residue converter, modular multipliers and modular adders. Therefore, a more systematic method is required to select a moduli set by considering the hardware complexity and delay of the complete RNS-based system.

*To whom all correspondence should be addressed.

In this paper, we shall discuss our moduli set selection methodology. The domain of application is restricted to FIR filters and filter banks. A software tool that assists system designers in moduli selection is implemented based on our methodology. This tool selects a moduli set based on the specifications of the filter bank, such as the number of taps, the required dynamic range and the type of filter bank. It also provides the designer with the estimated cost of the RNS-based designs in terms of area, delay and power consumption for the selected moduli set.

The rest of the paper is organized as follows. First, a brief review of residue number systems and moduli selection for RNS-based systems is provided in Section 2. Then, synthesis results of various parts for RNS-based FIR filter implementation are presented in Section 3. In Section 4, the design of the software tool is described, followed by some illustrative implementation results in Section 5. Lastly, conclusions are given in Section 6.

2. Preliminaries

2.1. Residue Number Systems

A residue number system is defined by a moduli set, which consists of n pairwise relatively prime integers $\{m_0, m_1, \ldots, m_{n-1}\}$. The useful computational range M of such a number system, which is called the legitimate range, is defined by the product of all moduli in the moduli set, i.e. $M = \prod_{i-0}^{n-1} m_i$. A residue number system with legitimate range M is able to uniquely represent unsigned numbers in the range of $[0, M-1]$, or signed numbers in the range of $[-\frac{M-1}{2}, \frac{M+1}{2} - 1]$ for odd M, and $[-\frac{M}{2}, \frac{M}{2} - 1]$ if M is even. These ranges are known as the dynamic ranges.

A number X within the dynamic range can be represented by the list of its residues with respect to the moduli defined in the moduli set.

$$X = \{x_0, x_1, \ldots, x_{n-1}\}$$

where for $i = 0, 1, \ldots, n-1$

$$x_i = \begin{cases} X \bmod m_i, & X \geq 0 \\ (M - X) \bmod m_i, & X < 0 \end{cases}$$

In RNS, addition, subtraction and multiplication can be performed entirely on the residue representation of the operands. If two numbers X and Y are represented as $\{x_0, x_1, \ldots, x_{n-1}\}$ and $\{y_0, y_1, \ldots, y_{n-1}\}$ respectively, then

$$|X \bigcirc Y|_M = \left\{ |x_0 \bigcirc y_0|_{m_0}, |x_1 \bigcirc y_1|_{m_1}, \ldots, |x_{n-1} \bigcirc y_{n-1}|_{m_{n-1}} \right\} \tag{1}$$

where the operator $\bigcirc$ can be addition, subtraction or multiplication.

Since numbers are commonly represented in two's complement form, conversions between residue and binary representation are required. In the forward conversion, the residue

with respect to modulus m_i of a number X, is obtained by

$$|X|_{M_i} = \left| \sum_{i-0}^{k-1} b_i |2^i|_{m_i} \right|_{m_i} \tag{2}$$

where $b_i \in \{0, 1\}$.

For residue to binary conversion, the Chinese Remainder Theorem (CRT) is commonly used. The CRT states that binary/decimal representation of a number in RNS form can be obtained from the equation (3) below, provided all moduli in the moduli set are pairwise relatively prime.

$$|X|_M = \{x_0, x_1, \ldots, x_{n-1}\}_{RNS} = \left| \sum_{i=0}^{n-1} \hat{M}_i \langle \alpha_i x_i \rangle_{m_i} \right|_M \tag{3}$$

where $M = \prod_{i=0}^{n-1} m_i$, $\hat{M}_i = M/m_i$, and $\alpha_i = |\hat{M}_i^{-1}|_{m_i}$ which is the multiplicative inverse of $\hat{M}_i$ with respect to m_i.

2.2. Moduli Set Selection

There are two common choices of moduli sets given a certain dynamic range. The first choice is to use a small number of moduli with of the form 2^n and $2^n \pm 1$. Some common examples are $\{2^n + 1, 2^n, 2^n - 1\}$, $\{2^n, 2^n - 1, 2^n - 1\}$, and $\{2^{n+1} - 1, 2^n, 2^n - 1, 2^{n-1} - 1\}$. The second choice is to use a larger set of moduli that is primarily made up of small prime numbers. {7, 8, 9, 11, 13, 17, 19} and {7, 11, 13, 15, 23, 29, 31} are good examples of this kind of moduli sets.

The first approach has the advantage of faster and simpler forward conversion and modular reduction in terms of delay and hardware complexity. This is because (1) for modulus of the form 2^n, conversion simply involves truncating unwanted bits; and (2) for moduli of the form $2^n \pm 1$, conversion can be performed by using a carry-save adder with carry end-around that results in minimum performance penalty. Due to the fact that the delay and complexity of the reverse converter depends on the number of moduli in a moduli set, this kind of moduli sets also results in simpler reverse converter implementation. For some moduli sets belong to this category, there exist very efficient reverse conversion methods. On the other hand, because the speed up is proportional to the size of a moduli set, this kind of moduli sets may not be able to provide sufficient speed up.

The second approach has the advantage of providing higher speedup because more moduli with short wordlengths are used, and probably more efficient use of lookup tables. However, the larger number of modulus in the moduli set results in a much more complicated reverse converter architecture. Furthermore, the reduction mechanisms are more complicated for these moduli.

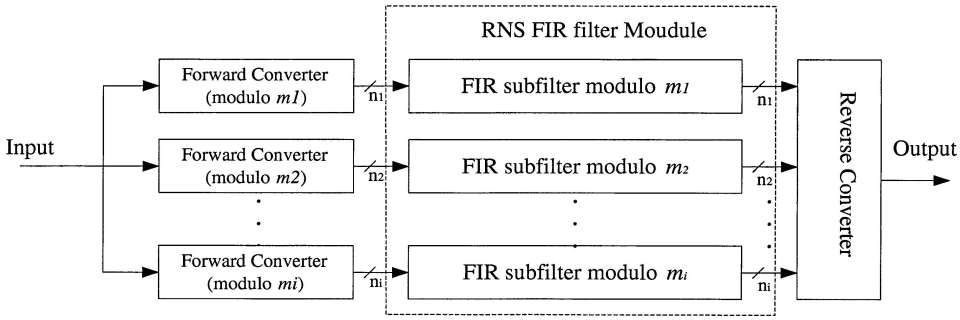

Figure 1. Structure of RNS-based FIR filter.

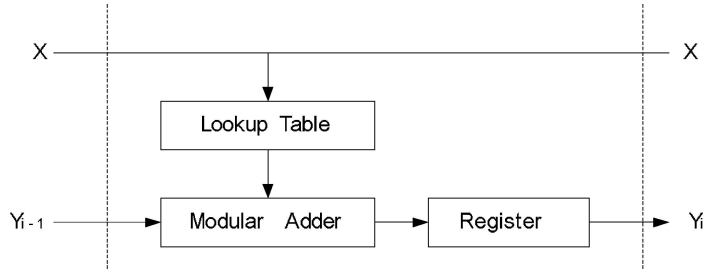

Figure 2. Tap structure of RNS FIR Sub-filter.

2.3. RNS-Based FIR Filter and Filter Banks

An RNS-based FIR filter usually consists of three parts: forward converters, RNS FIR filter module and the reverse converter as shown in Figure 1.

The FIR sub-filter can be implemented in direct form or transposed form. A tap structure for transposed form sub-filter is shown in Figure 2. Each filter tap has three basic components: a modular multiplier that performs the modular multiplication operation of input data and filter coefficients, a modular adder and the delay elements. In this design, the modular multipliers are implemented using the look-up table approach.

3. Hardware Cost Estimation

The cost of various parts of RNS-based FIR filter banks, such as forward converter, sub-filters and modular adders, are estimated and used as the basis for the moduli selection tool. Many architectures are available for forward and reverse converters and modular adders. In this particular design, for moduli that are in the form of $2^n \pm 1$, forward converter and modular adder architectures used are the ones proposed in [7] and [15] respectively.

For other general moduli, the forward converters are designed based on the multi-operand modular adder proposed in [6]. As for the modular adder, structure proposed in [1] is adopted.

Reverse converter structure depends on the actual moduli set used to implement the RNS-based filter. For some special moduli sets, such as $\{2^n + 1, 2^n, 2^n - 1\}$ and $\{2^n + 1, 2^n, 2^n - 1, 2^{n+1} - 1\}$, dedicated efficient reverse converter designs are available [2, 8, 12, 13]. While for other arbitrary moduli sets, a general CRT reverse converter design based on table look-up is used. The cost of a reverse converter is first estimated in terms of the number of adders and the size of lookup tables. And then the synthesis results for these individual components are used to calculate the total estimated cost of the reverse converters.

The estimated hardware cost is obtained from the synthesis results using Synopsys Design Analyzer with Avant! Passport 0.35 micron 3.3v optimum silicon technology library. Following design flow is adopted. First, various building blocks of an RNS-based FIR filter are coded using VHDL. Then, the forward converters, RNS-based sub-filters and reverse converters are constructed using those building blocks. After that, register transfer level (RTL) simulations are carried out to verify the correctness of various parts built in the previous step. After the correctness is verified, the designs are optimized by the synthesis process and the performances of various parts are evaluated. The area, delay and power estimation are obtained with clock frequency set to 200 MHz. The estimated cost for forward converters, RNS sub-filters and modular adders for different moduli with word length ranging from 2 bits to 11 bits are summarized in Tables 1, 2 and 3 respectively.

4. Software Tool for Moduli Selection and Cost Estimation

4.1. Overview

The proposed software tool is able to recommend a moduli set for implementing RNS-based FIR filter or FIR filter bank, based on a set of parameters, such as order of the filter and the required dynamic range. The estimate cost for the RNS-based FIR filter or filter bank is also generated based on the recommended moduli set.

This software tool can be roughly divided into three parts. First part is preprocessing and initialization module that preprocesses the input parameters and user requirements, and initializes the tool with the hardware cost of the forward converters, sub-filters and modular adders. Second part is the moduli set generation module. According to user requirements, it constructs and selects valid moduli sets with different numbers of moduli for comparison. The last part is the comparison module that compares the possible moduli sets generated from the second part, and chooses the moduli set that gives the lowest cost among all possible moduli sets to be the output of the tool, together with its cost in terms of area, delay and power consumption. The simplified overall flow diagram is shown in Figure 3.

Table 1. Estimated cost for forward converters

m_i	Area	Delay (ns)	Power (mW)
3	200	3.95	12.10
5	252	4.88	13.22
7	249	3.85	13.14
9	331	5.06	14.83
11	561	5.77	21.68
13	901	6.12	29.07
15	306	4.02	14.62
17	428	4.73	18.27
21	982	6.25	30.93
29	1100	7.25	36.00
31	309	3.43	16.13
33	431	4.33	19.76
63	409	3.51	18.78
65	465	4.95	19.86
127	431	3.81	18.47
129	530	4.57	21.10
255	512	3.60	20.45
257	558	4.20	22.11
511	616	3.80	22.65
513	623	4.70	23.30
1023	691	4.85	28.30
1025	710	5.30	31.30

4.2. Moduli Selection and Cost Estimation for RNS-Based FIR Filter

For the case of RNS-based FIR filter, user has to indicate the following inputs to the software tool: (1) the required dynamic range (number of bits); (2) filter's orders; (3) whether the filter coefficients are symmetrical; (4) the user requirements. The user requirements specify whether the design is time critical, area critical or power critical.

The software tool uses the user inputs as guild lines to construct valid moduli sets with m moduli, where $m \in Z, m \in [3, M]$, according to the following criteria: (1) $\prod_{i=0}^{m-1} m_i \geq 2^{DR-02}$, where DR is the required dynamic range specified by user. The dynamic range requirement has been relaxed a little bit, so that moduli sets with slightly less dynamic range will also be considered. (2) $GCD(m_i, m_j) = 1$ for $\forall i, j \in [0, m-1], i \neq j$, where $GCD(x, y)$ returns the greatest common divisor of integers x and y. These two criteria ensure that the constructed moduli sets have sufficient dynamic range and relatively prime. Hence, they are valid.

If a valid moduli set is found, the estimated cost of forward converters and RNS sub-filters for each modulus in the moduli set can be retrieved from the initialization data. Because the

Table 2. Estimated Cost of RNS Sub-filters

m_i	bits	Area	Delay (ns)	Power (mW)
3	2	34.55	2.64	2.30
4	2	22.61	1.66	1.63
5	3	39.96	2.86	3.06
7	3	56.16	3.09	3.13
8	3	37.47	2.05	2.51
9	4	79.68	3.84	4.16
11	4	130.95	4.70	6.84
13	4	126.48	4.67	6.78
15	4	75.07	3.52	5.54
16	4	50.19	2.65	4.44
17	5	96.83	4.32	6.46
21	5	209.3	4.68	11.16
29	5	171.10	4.66	10.59
31	5	98.80	3.98	7.09
32	5	85.89	2.65	6.16
33	6	118.92	4.66	8.01
63	6	119.89	4.42	8.51
64	6	107.88	3.20	8.04
65	7	147.69	4.87	11.33
127	7	142.60	4.65	9.86
128	7	130.90	3.33	9.26
129	8	180.71	5.09	12.35
255	8	181.09	4.79	11.16
256	8	146.25	3.59	10.62
257	9	221.21	5.48	14.07
511	9	222.48	4.79	14.63
512	9	183.30	3.75	13.82
513	10	258.80	5.72	16.55
1023	10	266.06	4.91	16.61
1024	10	253.55	3.96	15.23
1025	11	299.58	6.09	18.30

cost of the reverse converter can only be estimated by looking at the complete moduli set, the constructed moduli set is sent to a reverse converter cost estimation module to calculate the estimated cost of the reverse converter for this specific moduli set.

The moduli set that is most suitable for user requirements will be selected as the candidate set for moduli set with m moduli. By looking at all the candidate sets with different numbers of moduli, a final moduli set with lowest cost is selected according to the user requirements.

Table 3. Estimated Cost of Modular Adders

m_i	bit	Area	Delay (ns)	Power (mW)
3	2	34.55	2.64	2.30
4	2	22.61	1.66	1.63
5	3	39.96	2.86	3.06
7	3	56.16	3.09	3.13
8	3	37.47	2.05	2.51
9	4	79.68	3.84	4.16
11	4	130.95	4.70	6.84
13	4	126.48	4.67	6.78
15	4	75.07	3.52	5.54
16	4	50.19	2.65	4.44
17	5	96.83	4.32	6.46
21	5	209.3	4.68	11.16
29	5	171.10	4.66	10.59
31	5	98.80	3.98	7.09
32	5	85.89	2.65	6.16
33	6	118.92	4.66	8.01
63	6	119.89	4.42	8.51
64	6	107.88	3.20	8.04
65	7	147.69	4.87	11.33
127	7	142.60	4.65	9.86
128	7	130.90	3.33	9.26
129	8	180.71	5.09	12.35
255	8	181.09	4.79	11.16
256	8	146.25	3.59	10.62
257	9	221.21	5.48	14.07
511	9	222.48	4.79	14.63
512	9	183.30	3.75	13.82
513	10	258.80	5.72	16.55
1023	10	266.06	4.91	16.61
1024	10	253.55	3.96	15.23
1025	11	299.58	6.09	18.30

If user requires the design to be time critical, the tool will recommend a moduli set that has the shortest critical path delay. The critical path delay is calculated as the formula below:

$$CPD = \text{Max}(D(m_i)) + \text{Max}(FC\_D(m_i)) + RC\_D(\{m_i \mid i \in [0, m-1]\}),$$

where CPD denotes the critical path delay; $D(x_i)$ and $FC\_D(x_i)$ return the RNS sub-filter delay and the forward converter delay for a particular modulus x_i respectively;

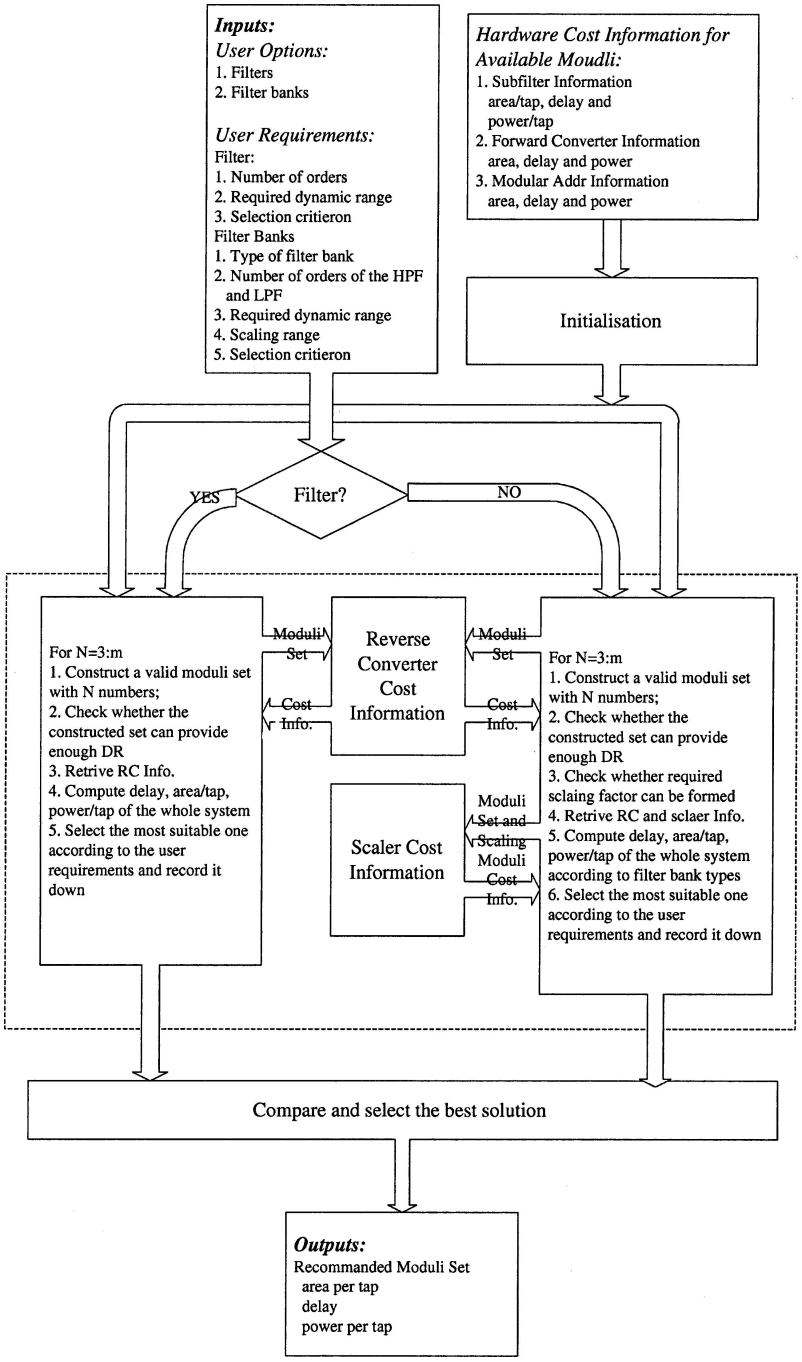

Figure 3. Program flow.

$RC\_D(\{x_1, x_2 \ldots x_m\})$ returns the reverse converter delay for a specific moduli set $\{x_1, x_2, \ldots, x_m\}$; the Max$(\cdot)$ function chooses the largest values returned by $D(x_i)$ and $FC\_D(x_i)$ which, together with the reverse converter delay, forms the critical path of an RNS-based FIR filter design.

If the design is required to be area critical or power critical, the selection of moduli set is on a '*per tap*' basis, which means that moduli set that occupies least area per filter tap or consumes least power per tap will be chosen to be the recommended moduli set.

The area occupied by an RNS-based sub-filter tap is calculated as

$$APT = \sum_{i=0}^{m-1} A(m_i) + \frac{\left(\sum_{i=0}^{m-1} FC\_A(m_i) + RC\_A(\{m_i \mid i \in [0, m-1]\}) \right)}{\text{taps}},$$

where APT denotes area per tap; $A(x_i)$ returns the area occupied per modulo x_i sub-filter tap; $FC\_A(x_i)$ returns the area occupied by a modulo x_i forward converter; $RC\_A(\{x_1, x_2 \ldots x_m\})$ returns the area occupied by the reverse converter for a specific moduli set defined by $\{x_1, x_2, \ldots, x_m\}$.

Similarly, the power consumption of an RNS-based sub-filter tap can be calculated as

$$PPT = \sum_{i=0}^{m-1} P(m_i) + \frac{\left(\sum_{i=0}^{m-1} FC\_P(m_i) + RC\_P(\{m_i \mid i \in [0, m-1]\}) \right)}{\text{taps}},$$

where PPT denotes power per tap; $P(x_i)$ returns the power consumed by a modulo x_i sub-filter tap; $FC\_P(x_i)$ returns the power consumed by a modulo x_i forward converter; $RC\_P(\{x_1, x_2, \ldots, x_m\})$ returns the power consumed by the reverse converter for a specific moduli set defined by $\{x_1, x_2, \ldots, x_m\}$.

The outputs of this software tool for RNS-based FIR filters include the recommended moduli set based on user inputs and requirements and the estimated cost in terms of area, delay and power consumption of the RNS-based FIR filter implemented using the recommended moduli set.

4.3. Moduli Selection and Cost Estimation for RNS-Based FIR Filter Banks and Multilevel Filter Banks

4.3.1. Word Length Reduction

Because no round off is allowed in RNS, moduli sets used to implement an RNS-based system must be able to provide sufficient dynamic range to accommodate results produced by the required arithmetic operations to avoid overflow. A simple dynamic range analysis shows that for multilevel filter banks, word length of intermediate outputs between filtering stages must be reduced. Otherwise, it is impractical to have a moduli set with very large dynamic range to implement the multilevel filter banks.

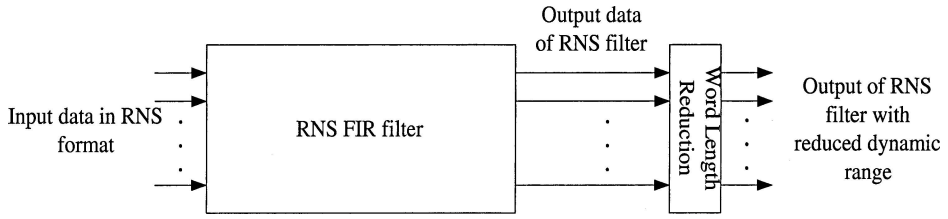

Figure 4. RNS-based FIR filtering stage.

In the software tool, we assume two approaches of word length reduction. The first approach is to use RNS scaling scheme proposed in [4], which uses look-up tables and modular adders. This approach scales the outputs with the product of some of moduli in a moduli set that approximately equals to the required scaling range. Another approach is to use the reverse conversion followed by forward conversion, which can produce exact scaling. The outputs are first converted back to binary domain through reverse conversion. Then the outputs are truncated in binary domain. After truncation, the outputs with reduced word length are then converted back to RNS domain through forward conversion. One RNS sub-filter and word length reduction unit form an RNS-based filtering stage as shown in Figure 4.

4.3.2. Types of Filter Bank Considered

In this software tool, three different types of tree-structured multilevel FIR filter banks are considered. A simplified illustration is shown in Figure 5. Figure 5(a) and (b) show 1-D and 2-D octave band tree-structured filter banks respectively. They are widely used to implement 1-D and 2-D discrete wavelet transforms. Figure 5(c) shows a tree-structure filter bank with equal pass bandwidth, which can be used to implement re-configurable channelizer for software defined radio receivers [11].

4.3.3. Program Design

For the case of RNS-based FIR filter bank or multilevel filter bank cost estimation, slightly more inputs are needed: (1) type of filter bank, which will be one of the three types defined in previous section; (2) low pass filter's orders; (3) high pass filter's order; (4) whether the filter coefficients are symmetrical; (5) whether the filter bank is a orthogonal or bi-orthogonal; (6) required dynamic range; (7) whether it is a analysis filter bank or a synthesis bank; (8) scaling range and (9) user requirements. Likewise, user requirements specify whether the design is time critical, area critical or power critical.

Program flow is similar to that of the RNS FIR filter case except that the way to construct moduli set is different and the estimated cost calculation is more complicated.

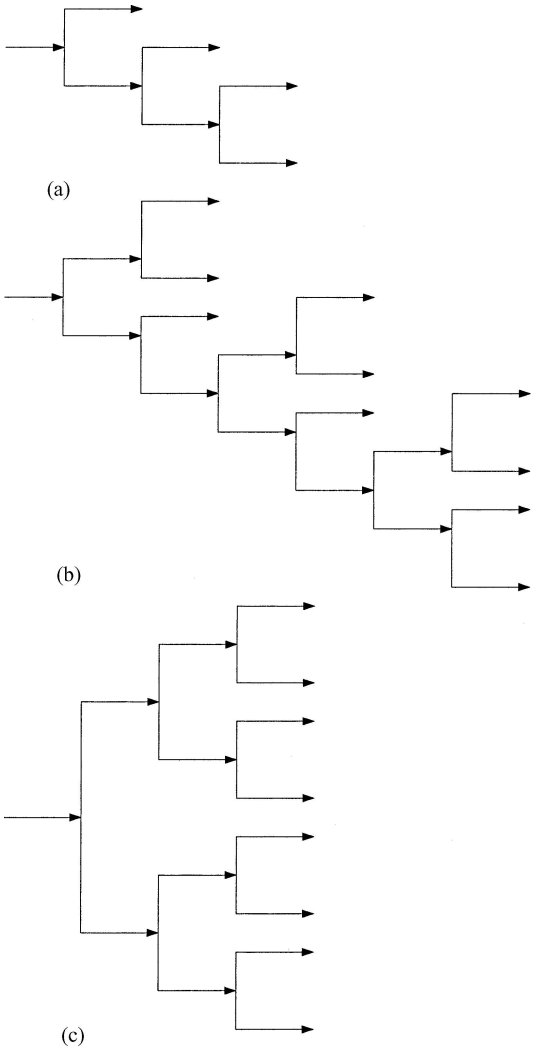

Figure 5. Different types of filter banks (Analysis).

If RNS scaling approach is used for word length reduction, moduli sets used for RNS-based FIR filter bank implementation have to fulfill the same requirements as that for RNS-based FIR filter. In addition, elements of the moduli set have to be able to facilitate the required scaling process, i.e. the product of some of moduli in a moduli set approximately equals to the required scaling range. For example, if the required dynamic range and the required scaling range are 32 bits and 12 bits respectively, moduli set I is defined as {127 65 64 33 31 7}, and moduli set II is defined as {1024 257 129 127}. Even though both moduli sets can satisfy the dynamic range requirement, moduli set I should be used

Table 4. Resources Requirements for Different Types of Filter Banks (Analysis)

	Forward converters	**RNS sub-filter taps**	**No. of word length reduction units**	**No. of Reverse converters**
Type1	1 set	$N(lp + hp)$	$N - 1$	2
Type2	1 set	$3N(lp + hp)$	$3N - 1$	4
Type3	1 set	$(2^N - 1)(lp + hp)$	$2(2^N - 1)$	2^N

because the product of some of its moduli approximately equals to the required scaling range ($65 \times 64 \approx 2^{12}$), which will facilitate the scaling process.

If, instead, the reverse conversion followed by forward conversion approach is used for word length reduction, the moduli set construction is the same as that of the RNS-based filter.

Resources requirements for RNS implementation of different types of N-level analysis filter banks are listed in Table. 4. The resources requirements of synthesis filter banks can be determined in a similar way. Costs in terms of area and power consumption can then be estimated based on the resources requirements.

In terms of critical path delay estimation, the following formulas are used for different types of analysis filter banks.
Type I:

$$\begin{aligned}&\text{Max}(D(m_i)) + \text{Max}(FC\_D(m_i)) + (N-1)SC\_D(\{m_i \mid i \in [0, m-1]|\}) \\ &\quad + RC\_D(\{m_i \mid i \in [0, m-1]|\})\end{aligned}$$

Type II:

$$\begin{aligned}&2\text{Max}(D(m_i)) + \text{Max}(FC\_D(m_i)) + (2N-1)SC\_D(\{m_i \mid i \in [0, m-1]|\}) \\ &\quad + RC\_D(\{m_i \mid i \in [0, m-1]|\})\end{aligned}$$

Type III:

$$\begin{aligned}&\text{Max}(D(m_i)) + \text{Max}(FC\_D(m_i)) + (N-1)SC\_D(\{m_i \mid i \in [0, m-1]\}) \\ &\quad + RC\_D(\{m_i \mid i \in [0, m-1]\})\end{aligned}$$

where $D(x_i)$ and $FC\_D(x_i)$ return the RNS sub-filter delay and the forward converter delay for a particular modulus x_i respectively; $SC\_D(\{x_1, x_2 \ldots x_m\})$ and $RC\_D(\{x_1, x_2 \ldots x_m\})$ return the word length reduction unit and reverse converter delay for a specific moduli set $\{x_1, x_2, \ldots, x_m\}$; the Max$(\cdot)$ function selects the largest values returned by $D(x_i)$ and $FC\_D(x_i)$. Likewise, the estimated critical path delay for synthesis filter bank can also be determined. The only difference is to consider the additional delay of the modular adders that merge data from the low pass and high pass branches.

Table 5. Recommended Moduli Sets and Estimated Cost for RNS-Based FIR Filters

Dynamic range		**20 bits**			**24 bits**			**32 bits**		
Filter taps		**Time critical**	**Area critical**	**Power critical**	**Time critical**	**Area critical**	**Power critical**	**Time critical**	**Area critical**	**Power critical**
10	Recommended moduli set	{129 128 127}	{128, 127, 63}	{128, 127, 63}	{257, 256, 255}	{33, 32, 31, 17, 7,5}	{127, 64, 63, 31}	{511, 257, 256, 255}	{65, 64, 31, 29, 17, 9, 7}	{127, 65, 64, 33, 31, 7}
	Delay (ns)	13.97	15.3	15.3	14.59	19.6	18.3	18.1	22.93	20.61
	Area/tap	1441.7	1187.6	1187.6	2080.3	1762.7	1948.9	3880.4	2904.7	2937.7
	Power/tap (mW)	31.2	28.8	28.8	36.97	39.77	36.7	56.56	61.19	56.03
100	Recommended moduli set	{129 128 127}	{32, 31, 17, 9, 7}	{31, 17, 9, 8, 7, 5}	{257, 256, 255}	{33, 32, 31, 17, 7, 5}	{31, 17, 5} 11, 9, 8, 7, 5}	{511, 257, 256, 255} 9, 7, 5}	{64, 31, 29, 17, 13,	{127, 65, 7} 64, 33, 31,
	Delay (ns)	13.97	19.77	19.77	14.59	19.6	20.5	18.1	23.5	20.61
	Area/tap	1241.8	846.6	874.48	1854.4	1081.9	1117.5	3092.1	1640.6	1706.3
	Power/tap (mW)	23.3	19.25	19.14	28.25	24.2	24.1	40.5	36.20	34.86
1800	Recommended moduli set	{129 128 127}	{32, 31, 17, 9, 7}	{31, 17, 9, 8, 7, 5}	{257, 256, 255}	{33, 32, 31, 17, 7, 5}	{31, 17, 11, 9, 8, 7, 5}	{511, 257, 256, 255}	{64, 31, 29, 17, 13, 9, 7, 5}	{129, 127, 64, 31, 17, 7}
	Delay/tap (ns)	13.97	19.77	19.77	14.59	19.6	20.5	18.1	23.5	20.64
	Area	1220.83	793.9	818.7	1830.7	1010.5	1040.9	3009.4	1505.9	1677.1
	Power/tap (mW)	22.53	18.89	17.7	27.33	22.56	22.26	38.82	33.29	32.64

Table 6. Recommended Moduli Sets and Estimated Cost for RNS-Based Filter Bank I

Filtre taps	Filter bank type and levels	Dynamic Range	24 bits		
			Time critical	**Area critical**	**Power critical**
LP = 9	Type = 2	Recommended	{257, 256, 255}	{32, 31, 17, 15,	{33, 32, 31, 17, 7, 5}
HP = 7	Level = 3	moduli set		11,7}	
		Delay (ns)	43.77	62.0	59.5
		Area/tap	2270.5	1935	1942.0
		Power/tap (mW)	62.45	55.68	54.24

The outputs of this software for RNS-based FIR filter banks include the recommended moduli set, moduli involving in the scaling process, if RNS scaling approach is used in word length reduction. Estimated costs of the RNS-based filter bank implemented using the recommended moduli set in terms of area, delay and power consumption are also provided.

5. Implementation Results

Some sample outputs of the proposed software tool are presented in this section.

Table 5 lists the recommended moduli sets and the estimated costs for RNS-based FIR filter implementation. Filters considered here have symmetric coefficients. For different required dynamic range and filter length, the tool recommends a moduli set that is most suitable for the user requirement.

Data in Table 5 show that trade-offs among delay, area and power consumption can be achieved by using different moduli sets during the RNS-based FIR filter implementation.

For the case of RNS-based filter bank implementation, two examples are considered. First one is a 3-level 2-D 9/7-tap analysis filter bank (Type II), which is widely used to implement 2-D discrete wavelet transform. Second example is a 9-level filter bank with equal pass bandwidth (Type III) that implements a dual mode channelizer for GSM and PCS [10]. The recommended moduli sets and the estimated costs for these two examples are listed in Tables 6 and 7 respectively.

6. Conclusions

The choice of moduli set is one of the most important issue in designing RNS-based FIR filters and filter banks. In this paper, we try to address this issue by designing a software tool for moduli set selection. Based on design inputs, such as number of filter taps, required dynamic range and filter bank types etc., the software tool is able to recommend a moduli set resulting in shortest delay or smallest area or least power consumption, according to the user requirement. The estimated cost in terms of delay, area and power consumption

Table 7. Recommended Moduli Sets and Estimated Cost for RNS-Based Filter Bank II

Filtre taps	Filter bank type and levels	Dynamic Range	20 bits		
			Time critical	Area critical	Power critical
LP = 1800	Type = 3	Recommended	{129, 128,	{32, 31, 17,	{31, 17, 9, 8, 7, 5}
HP = 1800	Level = 9	moduli set	127}	15, 7}	
		Delay/tap (ns)	257.45	388.02	393.1
		Area	725.7	492.36	508.1
		Power/tap (mW)	31.84	20.45	19.56

is also provided for reference. Outputs of the software tool also indicate that certain level of trade-off among delay, area and power consumption exists for the RNS-based filter and filter bank implementation by using different moduli sets.

References

1. Hiasat, A. A. High-Speed and Reduced Area Modular Adder Structures for RNS. *IEEE Trans. on Computers*, vol. 51, no. 1, Jan. 2002.
2. Hiasat, A. A. and H. S. Abdel-Aty-Zohdy. Residue-to-Binary Arithmetic Converter for the Moduli Set $2^k, 2^{k-1}, 2^{k-1} - 1$. *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 45, no.2, Feb. 1998.
3. Jenkins, W. K., and B. J. Leon. The Use of Residue Number Systems in the Design of Finite Impulse Response Digital Filters. *IEEE Trans. on Circuits and Systems*, vol. CAS-24, pp. 191–201, 1977.
4. Jullien, G. A. Residue Number Scaling and Other Operations Using ROM Arrays. *IEEE Trans. On Computers*, vol. C-27, pp. 325–326, Apr. 1978.
5. Kaluri, K., W. F. Leong, K.-H. Tan, L. Johnson, and M. Soderstrand. Comparison of RNS and Optimized FIR Digital Filters in Xilinx FPGA's. In *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems*, 2001, vol. 1, pp. 438–441.
6. Piestrak, S. J. Design of Residue Generators and Multioperand Modular Adders Using Carry-Save Adders. *IEEE Trans. on Computers*, vol. 423, no. 1. Jan. 1994.
7. Pourbigharaz, F., and H. M. Yassine. Simple Binary to Residue Transformation with Respect to $2^{m+}1$ Moduli. *IEE Proceedings on Circuits Devices and Systems,* vol. 141, no. 6, pp. 522–526, 1994.
8. Premkumar, A. B. An RNS to Binary Converter in $2n + 1$, $2n$, $2n - 1$ Moduli Set. *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, no. 7, pp. 480–482, 1992.
9. Soderstrand, M. A., W. K. Jenkins, G. A. Jullien, and F. J. Taylor (Eds), *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*. IEEE Press, New York, 1986.
10. Soudris, D., K. Sgouropoulos, K. Tatas, V. Pavlidis, and A. Thanailakis, A methodology for implementing FIR filters and CAD tool development for design RNS-based systems. In *Proceedings of International Symposium on Circuits and Systems*, 2003.
11. Vinod, A. P., E. M.-K. Lai, A. B. Premkumar, and C. T. Lau. A reconfigurable multi-standard channelizer using QMF trees for software radio receivers. In *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,* Beijing, China, 2003.
12. Vinod, A. P., and A. B. Premkumar. A Memoryless Reverse Converter for The 4-Moduli Super Set $\{2^n + 1, 2^n, 2^n - 1, 2^{n+1} - 1\}$. *Journal of Circuit, Systems and Computers*, vol. 2, nos. 1/2, 2000.

13. Wang, Y., X. Song, M. Aboulhamid, and H. Shen. Adder Based Residue to Binary Number Converters for $(2^{n+1}, 2^n, 2^n - 1)$. *IEEE Trans. on Signal Processing*, vol. 50, no. 7, pp. 1772–1779, 2002.
14. Wang, W., M. N. S. Swamy, and M. O. Ahmad. Moduli selection in RNS for efficient VLSI implementation. In *Proceedings of International Symposium on Circuits and Systems*, 2003.
15. Zimmermann, R. Efficient VLSI Implementation of Modulo $(2^k \pm 1)$ Addition and Multiplication. In *Proceedings of 14th IEEE Symposium on Computer Arithmetic,* Adelaide, Australia, Apr., 1999, pp. 158–167.