Bit-Serial, VLSI Architecture for the Implementation of Maximum-Length Number-Theoretic Transforms Using Mixed Basis Representations

M.G.Parker,M.Benaissa

School of Eng., Univ.of Huddersfield, Huddersfield HD1 3DH, UK.

Abstract

Fermat and Mersenne NTTs are relatively easy to implement, but unsuitable for many DSP applications, due to small block length over wordlength. This paper presents VLSI design techniques appropriate for a wider range of NTTs, including maximum-length NTTs, and presents a systolic architecture exploiting block-length factorisation to decompose the architecture into sub-modules, themselves systolic NTTs. The design avoids the explicit implementation of modular addition, accumulation and multiplication by using systolic basis converters which, inherently, perform the aforementioned tasks. The implementation of a maximum-length, 60pt NTT is described as an example. It uses binary to ternary basis conversion to perform all addition and multiplication, and 'ternary basis compression' to perform accumulation.

1 Introduction

A general 1-D NTT of a sequence x[n] is defined as,

$$X[k] = \left\langle \sum_{n=0}^{N-1} x[n].p^{nk} \right\rangle_M \qquad 0 \le n,k \le N-1 \quad (1)$$

where p has order N mod M, $\langle N^{-1} \rangle_M$ exists and is unique, and an N-point NTT can be defined over all the moduli factors of M [1]. By choosing all NTTs with 2 as the kernel, over a modulus, M, where $\langle 2^N \rangle_M = 1$, a wide range of block lengths, for a given dynamic range, become available [2,3]. As with FNTs and MNTs, multiplication can be implemented as bit shifts but, in general, the problem of modulus overflow is non-trivial. This paper shows how NTT block length factorisation enables a regular and highly-pipelined decomposition. Basis converters [4] distribute the addition over a series of simple cells, whilst converting the data basis to a form which eliminates multiplication. Finally, the concepts are illustrated by considering the specific implementation of a 60pt NTT, mod 61, using binary and ternary representations.

2 Factorisation of NTTs

An N-length NTT is factorised as follows,

$$X[k] = X_\mu[\theta] \text{ for } k = N_0.\theta + \mu$$

where $0 \le \theta < N_1$, $0 \le k < N$,
$0 \le \mu < N_0$ and $N = N_0.N_1$,

and,

$$X_\mu[\theta] = \left\langle \sum_{n_0=0}^{N_0-1} \sum_{n_1=0}^{N_1-1} g^{n_1.\theta} . \left[g^{D_{N_0}((n_1+N_1.n_0).\mu)} . p^{\langle (n_1+N_1.n_0).\mu \rangle_{N_0}} . x[n_1 + N_1.n_0] \right] \right\rangle_M$$

where $g = \langle p^{N_0} \rangle_M$, $D_z() = ()$ DIV z, and
$n = n_1 + N_1.n_0$ where,
$0 \le n_0 < N_0, 0 \le n_1 < N_1$

We have expressed $X_\mu[\theta]$ as a summation of N_0 seperate N_1pt NTTs, where $x[n]$ is pre-processed to become,

$$g^{D_{N_0}((n_1+N_1.n_0).\mu)} . p^{\langle (n_1+N_1.n_0).\mu \rangle_{N_0}} . x[n_1 + N_1.n_0]$$

The architectures, developed, perform the pre-processing by pipelining the small powers of p multiplications, $(p^0 \; . \; . \; p^{N_0-1})$, followed by the small power of g multiplications, (implemented as a modification to the subsequent N_1pt NTT architectures).

Fig 1 shows the realisation of this decomposition. Gate rotators re-route the stream of partial

0-7803-0946-4/93 $3.00 © 1993 IEEE

products, for successive $x[n]$, between each processing stage and are general to any block length. Fig 1 also shows the equivalent, unfactorised, architecture. The design combines N_0, N_1pt, NTTs, together with a pre-processing stage, to form the complete NTT, and, if N_1 is, itself, factorisable, the decomposition is repeated for each N_1pt NTT.

The N_1pt NTT modification, necessary for the extra small powers of g multiplications, manifests itself as a series of cumulative pulses to the GRN_1 modules. These pulses update the re-routing pattern within the rotators accordingly.

3 NTT Multiplications Using Basis Conversion

To simplify designs, we assume α, the i/p basis parameter (see appendix) $= p$, the kernel of the transform. (Binary input data $\Rightarrow \alpha = p = 2$). Thus small power of p multiplications become a few additions, mod M. For instance, if $\langle p^{N_h} \rangle_M = h = p^0 + p^1$, multiplications by $p^0 \; . \; p^{N_h - 1}$ become single additions of bit re-orderings of $x[n]$. As shown in the previous section, we require multiplications by $p^0 \; . \; p^{N_0 - 1}$. These multiplications become single additions if $N_h \geq N_0$. Ideally $N_h = N_0$. These additions can be incorporated into the basis conversion process.

The converter changes the data from a p-basis to a g-basis to simplify multiplication by small powers of g. g is the kernel of the N_1pt NTTs and, as $\langle g^{N_1} \rangle_M = 1$, power of g multiplications become bit re-orderings.

Each converter cell contains a number of digits, ('bits' for binary digits), which are related by a state transition table and identical cells are 'joined' together to form the complete systolic converter.

If N_1 is, itself, factored, the same process is repeated and g-basis to 'power of g'-basis converters will be required, (and so on).

4 Compression Replaces Accumulation

Each accumulator sums N data words over a g-basis to calculate $X[k]$. By interpreting these N serial data words as one long data word with an identical basis flow, (apart from $j'_N = N.j'$, where j'_N refers to the long word) (see appendix), this single data word $= X[k - z]$, (where $g^{j'} = p^z \Rightarrow z = N_0.j'$). As the mapping from $X[k]$ to $X[k - z]$ is one-to-one, accumulation is not strictly necessary, and for many applications, (such as convolution), accumulators can be omitted from the design, giving substantial hardware savings.

If it is desired to reduce the value of j'_N to a minimum, basis compressors, similar in design to the basis converter, but with minimal feedback inserted between cells, are used.

5 Optimal NTTs

The design principles, developed in previous sections, are applicable to any NTT. One class of NTTs for which they are particularly suited is maximum-length NTTs. These occur when M is prime and $N = M - 1$ (eqn 1). Table 1 list a few such NTTs for $p = 2$. When N is factorisable, a suitable $N_h \geq N_0$ should be found so that h is a simple additive combination of small powers of 2, where $h = \langle 2^{N_h} \rangle_M$. As explained in the last section, these simple additions can then be incorporated into the basis conversion where the basis converters are required to convert from a binary basis to a g-basis. Suitable N_h, N_0,h and g are given in Table 1 for each example, along with the expression of h as a simple modular addition.

M	N	N_h	N_0	h	h_{Add}	g
11	10	8	5	3	$2^0 + 2^1$	32
13	12	9	6	5	$2^0 + 2^2$	64
29	28	5	4	3	$2^0 + 2^1$	16
61	60	6	6	3	$2^0 + 2^1$	3
4093	4092	12	12	3	$2^0 + 2^1$	3

Table 1: Possible First-Stage Factorisation of Selected Prime Moduli

6 High Thr'put VLSI Implementation of a 60pt NTT, Mod 61

This example of a maximum-length NTT implements eqn 1 when $p = 2$, $M = 61$ and $N = 60$, and is shown in Fig 2. The input basis flow is given by,

$$\leftarrow 2^5, 2^4, 2^3, 2^2, 2^1, 2^0$$

where $j = j' = 6$ and $d = 1$,

and, after binary to ternary conversion, by,

$$\leftarrow \begin{array}{ccccc} 3^4, & 3^3, & 3^2, & 3^1, & 3^0 \\ 3^9, & 3^8, & 3^7, & 3^6, & 3^5 \end{array}$$

where $\langle 3^{10} \rangle_{61} = 1$, $j = 5$, $j' = 6$ and $d = 2$.

Both bases span the data range $0 - 60$ and are therefore acceptable.

We factorise $N = 60$ as $2.3.5.2 = N_0.N_1.N_2.N_3$. As $\langle 2^6 \rangle_{61} = 3 = 2^0 + 2^1$, multiplications by

2^0,2^1,2^2,2^3,2^4,2^5 are implemented as single additions of bit re-orderings of $x[n]$. These re-orderings can be factorised into 2^0,2^1, followed by 4^0,4^1,4^2 multiplications. The additions are incorporated into the basis conversion. The 10pt NTTs, $(N_2.N_3)$, use a kernel, g, $= \langle 2^6 \rangle_{61} = 3$. The basis flow, generated by the converter, contains all possible powers of 3, mod 61. Thus, multiplications by 3^0,3^1,3^2,3^3,3^4 become bit re-orderings which are performed on output from the converters. Finally, multiplications by -1^0,-1^1 are achieved by re-routing the two basis flow lines of the ternary basis, as $\langle 3^5 \rangle_{61} = -1$,

By interpreting the convertor output as one, long, word, the output becomes $X[k-z]$ instead of $X[k]$, where $\langle 3^6 \rangle_{61} = \langle 2^z \rangle_{61} \Rightarrow z = 36$.

Should basis compression be desired, a basis converter architecture with feedback can be used.

The complete NTT design has been successfully simulated down to the logic level and is currently being built.

7 Conclusion

In this paper, a systolic architecture for a wide range of NTTs, especially maximum-length NTTs, has been developed. This architecture is based on the block-length decomposition of the NTT and the use of mixed-basis arithmetic. This allows efficient and more practical NTT-based systems to be implemented in VLSI.

8 Appendix - 'Basis' and 'Basis Flow'

'Basis' refers to the weightings given to a group of digits which, when evaluated and summed, represent a number. An α-basis means all weightings are successive powers of α. In this paper, each digit is a binary digit, ('bit'), having two values, $0, 1$, although the theory can be extended to any digit range. Ternary, in this context, implies $\alpha = 3$, whilst each digit is a 'bit'.

'Basis Flow' refers to the flow characteristics of the basis data and the weighting pattern is shown below for one word, as a $j' * d$ vector.

$$\begin{array}{cccccc} W_0.\alpha^{j-1} & W_0.\alpha^{j-2} & \cdots & W_0.\alpha^0 & 0 \cdots & 0 \\ W_1.\alpha^{j-1} & W_1.\alpha^{j-2} & \cdots & W_1.\alpha^0 & 0 \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ W_{d-1}.\alpha^{j-1} & W_{d-1}.\alpha^{j-2} & \cdots & W_{d-1}.\alpha^0 & 0 \cdots & 0 \\ \leftarrow & \multicolumn{2}{c}{j \text{ clock cycles}} & \rightarrow & & \\ \leftarrow & \multicolumn{4}{c}{j' \text{ clock cycles}} & \rightarrow \end{array}$$

where the vector is assumed to be travelling right to left. j' is the word period and $j' - j$ is the inter-word delay, both in clock cycles and d is the data path width. With no delay between successive words, $j' = j$ and when $d = 1$, the data moves in a digit-serial fashion.

References

[1] J.H.McClellan,C.M.Rader, **Number Theory in Digital Signal Processing,** Prentice Hall, '79

[2] O.R.Hinton, "Implementation and Application of Generalised Number-Theoretic Transforms," *IEE Colloqium on Signal Processing Applications of Finite Field Mathematics*, June '89

[3] H.Lu,S.C.Lee, "A New Approach to Solve the Sequence-Length Constraint Problem in Circular Convolution Using Number-Theoretic Transform," *IEEE Trans on Signal Processing*, Vol 39, No 6, pp 1314-1321, '91

[4] B.Parhami, "Systolic Number Radix Converters", *The Computer Journal*, Vol 35, No 4, '92

Fig 1 Architecture
For an Npt NTT
Mod M, Where
$N = N_0.N_1$

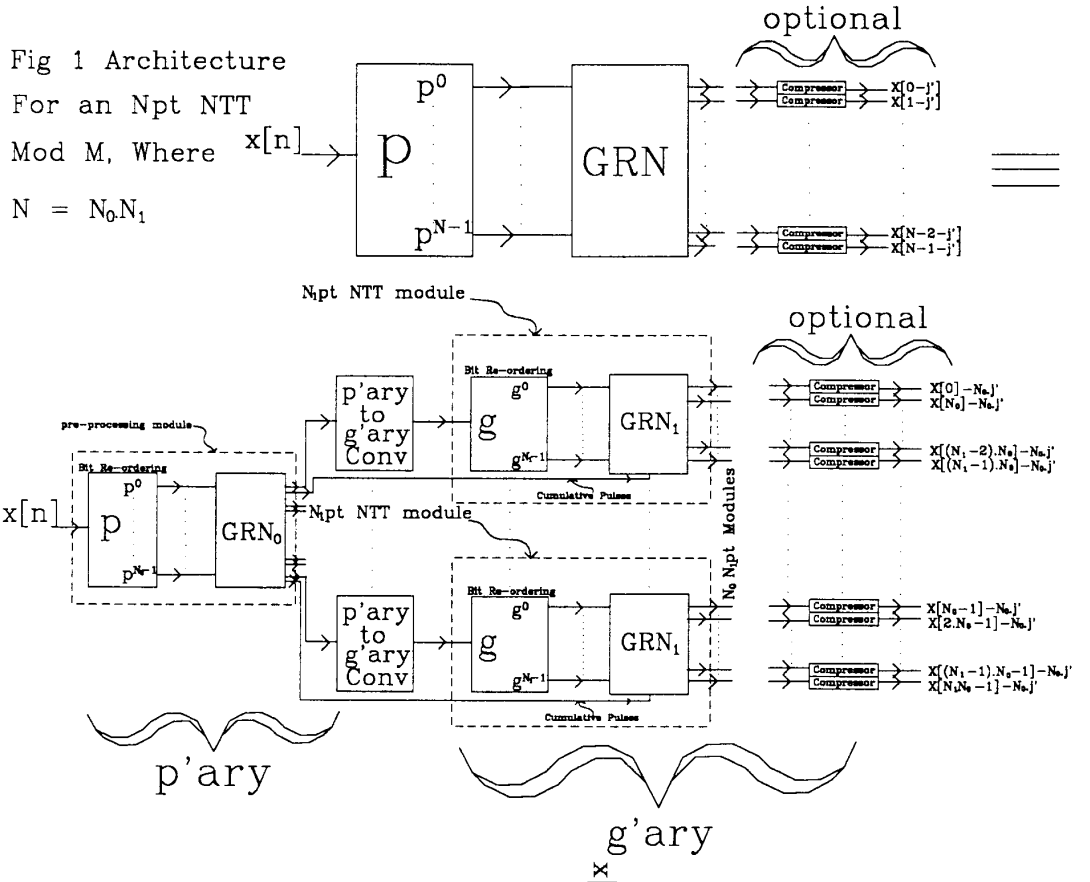
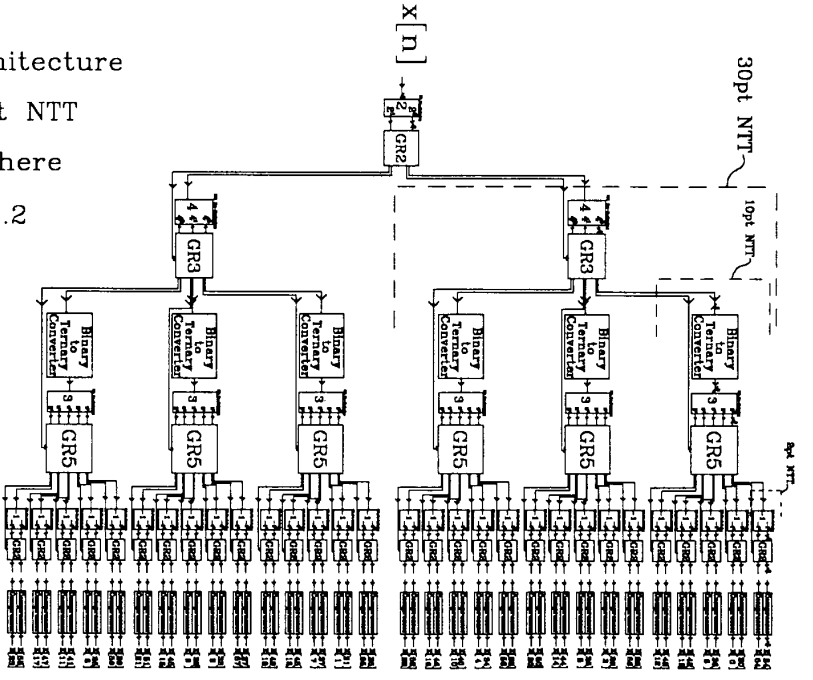
Fig 2 Architecture
For a 60pt NTT
Mod 61, Where
$60 = 2.3.5.2$