NUREG/CR-4170

# An Ultra-High Speed Residue Processor for SAFT Inspection System Image Enhancement

Final Report
October 1983 through March 1984

Prepared by J. N. Polky, D. D. Miller

Sigma Research Inc.

Prepared for
U.S. Nuclear Regulatory
Commission

GPO Printed copy price. $3.75

NUREG/CR-4170
R5

# An Ultra-High Speed Residue Processor for SAFT Inspection System Image Enhancement

Final Report
October 1983 through March 1984

Manuscript Completed: May 1984
Date Published: March 1985

Prepared by
J. N. Polky, D. D. Miller*

Sigma Research, Inc.
565 Industry Drive
Seattle, WA 98188

*BAE Automated Systems

**Prepared for**
**Division of Engineering**
**Office of Nuclear Regulatory Research**
**U.S. Nuclear Regulatory Commission**
**Washington, D.C. 20555**
**NRC FIN D1133**
**Under Contract No. NRC-04-83-186**

**ABSTRACT**

The Phase-I feasibility study of residue number system (RNS) image processing for SAFT inspection has successfully determined that an advanced inspection system may be built using a correlation-reconstruction SAFT algorithm, implemented with RNS techniques and off-the-shelf electronic components. Images are reconstructed in a number theoretic transform domain with simple pointwise multiplication of the A-scan data volume by a custom point spread function (PSF), all in a highly parallel computational architecture. These methods also allow image enhancement to be easily performed for improved flaw visualization, and with negligible speed reduction. It has been determined that high resolution three dimensional flaw images may be generated and that a commercially viable product could result through development of a prototype real-time RNS processor. The hardware is expected to be made up of 100 nsec bit slice microprocessor components and large RAM storage units. Such a processor would provide the NDE community with a valuable new tool, that could generate significant improvements in real-time flaw visualization of critical components encountered in nuclear power plant inspection. Special inspection conditions, such as multiple reflections and crack tip detection in the presence of corner reflecting effects, may be overcome by modifying the functional form of the PSF. It is also possible that semi-automatic flaw classification could be included in the process without significantly adding to the computational burden. Based on the performance estimates of the Phase-I effort, this new image processing system has the potential to acquire and focus the equivalent of 145 A-scans per second, which translates into more than 1000 cubic inches per min. inspection rate for typical pressure vessel specimens. A near-term prototype version can be fabricated, using some disk memory to reduce risk and cost, with a throughput of 40 focused A-scans per second. This prototype design may be easily upgraded to the higher throughput rate for a future production model.

**CONTENTS**

# 1.0 INTRODUCTION

The complex nature of nuclear power plant inspection puts severe requirements on nondestructive evaluation techniques. Not only is there a large mass of material, which must be tested in a reasonable time period, but the three dimensional characteristic of component flaws puts an almost impossible burden on existing data processing systems. Human operators of test instruments may be exposed to a high backround radiation environment, also making inspection speed important.

The large data volumes associated with nuclear plant inspection are comparable to other compute bound problems, such as weather modeling, geophysical exploration, aerodynamic research and fusion research. The algorithms which process these data sets are usually configured to produce solutions in realistic time periods, but without so much simplification that unrealistic results are generated. For most inspection situations, the data volume is not so large that it is impossible to handle; but it is considerably larger than most ultrasonic inspection systems are designed to handle. For example, many systems designed to evaluate small rods or disks at high resolution will collect data from only a few cubic centimeters, but nuclear reactor systems may require inspection of several cubic meters of material.

Research concerning the stucture and solution of compute bound problems has produced processing speed gains by either optimizing existing algorithms or by generating new algorithms with much more efficient structures. This type of improvement usually occurs as a first order enhancement step; with only limited long term success. Another method which falls into a similar category with algorithm modification is hardware speed up. This might involve a trivial replacement of processor hardware; an extreme example would be a pocket calculator replaced by a CRAY-1 (with a not so trivial price tag). A more practical, and currently very popular, solution is to

use a distributed or multi-processor architecture; involving parallel processing of the data set with many relatively small hardware units. Because these hardware modules are all acting in unison, the processing throughput can increase by a factor proportional to the number of units used. In many applications a distributed architecture can provide the required speed enhancement, but in some cases the algorithms may be very difficult to partition. Nearly all parallel processing schemes gain speed by increasing throughput, and correspondingly increased hardware. Residue number system methods (Appendix A) offer a different approach to parallelism by introducing it at the basic computational level, which is reflected in a top to bottom, parallel architecture. The main advantage of the residue number system is the increased computational speed due to its inherent parallel nature. Upon encoding the input data to residue form, the computation simultaneously takes place in a number of independent hardware units with no carries present. Hence, an arithmetic unit performs a full arithmetic operation in one clock period. This can be contrasted with the parallelism of a pipelined adder or multiplier of a binary vector computer. For the latter, once the pipe is filled, a new result is obtained each clock cycle, but a number of clock cycles are required from initiation to completion of the operation (the latency of the operation). The RNS achieves increased speed through arithmetic parallelism without introducing parallelism into the algorithms, thus providing a natural speed-up of scalar applications. Other advantages of the RNS include the ease of very high precision computation and error detection and correction.

In addition to its potential for very high speed computing, the RNS is attractive due to the architectural simplicity it offers. Processors can generally be fabricated from discrete, off-the-shelf components with data flow paths arranged in a staightforward and highly regular manner. As opposed to a von Neumann architecture in which complex data routings must be configured between low and high

speed memories and a single arithmetic unit, the RNS naturally lends itself to the use of many arithmetic elements and simple data management schemes; pipelining, for example, is easily implemented in one clock cycle. In the binary system where multiplication is much slower than addition, pipelining is achieved with delays and more complex clock circuits. Current research indicates that such an approach to solving compute bound problems can be effective for speed enhancement and its generic parallelism can allow simpler fabrication procedures, a less complex control structure, and greater computational precision than other methods. An RNS architecture can also allow the use of more sophisticated computational algorithms, which can produce superior results when applied to image processing problems encountered in nuclear power plant NDE.

Current regulations require detailed inspection of critical plant components, such as: circumferential and longitudinal welds on pressure vessel walls, all welds on vessel piping, and nozzels, and where pipes join vessel walls. In all cases the flaw throughwall dimensions and length must be quantified by the inspection system in order for fracture mechanic analysis to accurately determine critical size proximity. Therefore, an advanced flaw sizing system must be able to produce high resolution, three-dimensional images of a suspect component volume. Many previously developed ultrasonic devices are unsuitable for such a system primarily because they only generate two-dimensional B-scan images, derived from a one-dimensional aperture. In order to produce high resolution flaw images, it is absolutely necessary that an imaging system incorporate all three degrees of freedom. There are several advanced systems under development which propose to satisfy this requirement; they include, ALOK, acoustic holography, and coherent-sum SAFT. Of all these methods, the coherent-sum SAFT has the potential for the greatest improvement in inspection technology, primarily, because of its ability to give increased lateral resolution as well as superior depth resolution. The

relatively large synthetic aperture affects the lateral dimension while broadband acoustic excitation is responsible for depth resolution which is better than narrowband excited holographic methods. SAFT appears to be the best choice for a real-time inspection system applied to in-service evaluation of nuclear power plant components.

## 2.0 Phase-I Goals

It was determined, at the beginning of Phase-I, that the complete SAFT process should be evaluated, rather than our originally proposed study of only an image enhancement processor. The intention of this redirection was to provide the NRC technical monitor with an understanding of the stand alone capability of real-time RNS data processing as applied to NDE systems. The basic approach, as stated in our Phase-I proposal, was still valid for this expanded effort: algorithm selection and reformulation into an RNS compatible form; design, analysis, and performance evaluation of the computational process; and processor performance evaluation of a conceptual SAFT/RNS architecture. In addition, it was expected that a better understanding of the development feasibility, production cost and benefits to NDE would result from this effort.

An important motivating factor for this redirection was the realization that if RNS methods turned out to be usefull, then it would be better to have designed a complete RNS processor rather than compromise its performance by modifying the computational architecture in order to fit into an existing system. Furthermore, it was determined that the Coherent-Sum SAFT algorithm was not a good match with RNS methods due to its simple arithmetic structure and intense memory management requirements. For conventional binary arithmetic, such a system is very efficient, but RNS methods perform best in a computationally intense environment. Therefore, new goals were established to better exploit the potential speed advantage of RNS computational architectures. These include:

1) Find an RNS compatible algorithm for flaw imaging, which has a simple data/memory management method and has a regular computational structure at the expense of adding more arithmetic complexity.
2) Determine if image enhancement functions can be integrated into the reconstruction process, for further speed improvement.

## 3.0 Algorithm Selection

As a result of our investigation of complex image reconstruction techniques, it was determined that both goals could be realized by generating direct three dimensional reconstructions through the use of a correlation technique (Norton-1976). A correlation algorithm has the effect of greatly increasing the number of multiplies and adds, but it is efficiently implemented with RNS techniques.

In recent years the RNS has been studied for the high speed and hardware simplicity it offers to special purpose digital processing systems. A number of efforts have been directed toward the design of RNS architectures in order to perform fast Fourier transforms (Tseng-1979), number theoretic transforms and Fourier based convolutions and correlations (Jullien-1981), and the Good-Winograd Fourier transform (Taylor and Huang-1981). Efforts have concentrated on these transform algorithms because exceedingly high data throughputs are required for applications such as synthetic aperture radar digital image reconstruction.

Significant speed improvement can be gained when an RNS number theoretic transform (NTT) method is also used (Appendix B). The NTT is analogous to the discrete Fourier transform (DFT) where the complex number field is replaced by a finite field or a product of finite fields. The twiddle factors of the DFT, EXP(-2ik/N), k=0..N-1 are replaced with the powers of an Nth root of unity in

the field. There are two principle advantages of the NTT over the FFT: first, all arithmetic is exact with no roundoff error, and second, the twiddle factors do not have to be scaled and quantized to integers. Hence, the intermediate computed values do not grow unmanageably in the NTT and intermediate scaling (a difficult operation in the RNS) can be avoided.

The coherent-sum SAFT algorithm is fundamentally a three-dimensional correlation process, implemented as a series of one-dimensional correlations between a delay table and each A-scan. The delay table can also be thought of as a four-dimensional point-spread-function (PSF). Therefore, coherent-sum SAFT represents a subset of the more general Green´s function technique, commonly used to solve the acoustic ( also eletromagnetic ) wave equation for the three-dimensional broadband reflectivity function. In most forms of this method various assumptions are made in order to simplify the solution procedure. These usually include such techniques as far-field approximations (Fraunhofer or Fresnel), weak reflection approximations (Born), and various geometric assumptions. In most cases, a researcher´s goal is to develope a closed form analytical solution where these approximations are necessary, because of the complex nature of the wave solution. When real-time processing is required, too many simplifications could produce a severely degraded output. Fortunately, the formulation of coherent-sum SAFT yields a very simple numerical procedure for forming images from A-scan data. Previously it was thought that although the SAFT method is conceptually simple, the implementation of it in a real-time process would suffer from a fatal bottleneck: the delay calculations, involving many square root operations. By assuming a simple (i.e.planar) and uniform geometry the delay tables could be precomputed and used repeatedly. Current

development of a coherent-sum SAFT process employs delay tables, an optimized algorithm and a multi-processor architecture, with the hope of producing a real-time (i.e. scanner limited) imaging system.

The coherent-sum SAFT method will suffer from inaccurate reconstruction when wide-band excitation is used. The reconstruction process (calculation, or look-up, of the delay time from each sensor position to a single pixel location; selection of the corresponding A-scan data value, based on each delay value; adding the A-scan values together) assumes that target return echoes will constructively add, while nontarget values will be out of phase and destructively add. The underlying assumption of a temporally coherent source is only valid for monochromatic excitation. When wide-band or short duration pulses are used, a relatively poor quality, low resolution image will result. In practice, the R.F. signal will be partially coherent due to a temporally extended ringdown in the transmitted impulse. In this case the image will represent a suboptimal reconstruction in terms of signal to noise ratio and it will exhibit a reduction in depth resolution.

Although the current development architecture is expected to be a multi-processor type, its ultimate speed will be limited by its use of conventional binary arithmetic, and relatively conventional hardware techniques. The serial relationship between image construction and image enhancement will severely limit the overall inspection time, and there seems to be no easy way to integrate these functions into the coherent-sum algorithm. Finally, when special inspection conditions occur, such as multiple bounce and crack tip detection with corner reflecting effects, then the generic inflexibility of the coherent-sum SAFT algorithm will become a problem. Clearly, the current development prototype will advance the state of the art in in-service inspection technology, but only as a short term solution to an extremely complex flaw

visualization problem. An advanced processor design, such as a correlation-reconstruction SAFT algorithm, implemented with an RNS architecture could produce enhanced flaw imaging capability with a strong potential for growth into an automatic flaw classification and identification system.

## 4.0 The Correlation-Reconstruction Algorithm

The correlation-reconstruction method fundamentally consists of convolving the acoustic impulse response function for the test volume with the A-scan aperture data function to realize a focused volume image of the scattering function. Depending on the relative sizes of these functions, either a direct convolution or a Fourier transform method may be selected for the computational process. Practically speaking, the point-spread-function (PSF) for an acoustic source below a plane interface, measured by scanning along the surface, has a hyperbolic shape which varies with depth. Therefore, a different PSF would be needed for each depth slice in the correlation process. This space-variant effect is similar to the depth dependent delay function table which is calculated in the coherent-sum algorithm. It has been shown by Norton, Fig.1, that by introducing a substitution of $Z'=Z^2$ and $rho=R^2$ for both the data volume and the PSF volume, a spatially invariant function may be generated. Therefore the SAFT process can be efficiently implemented as a 3-D correlation between the data set and the modified PSF, all in a large dynamic memory without the need for significant memory management processes.

There is a slight practical problem with the geometric transformation to spatial invariance, and that is memory overflow. This would occur since the procedure would involve repositioning the original data and filling in open spaces into an expanded memory space by an index squaring process. For example, a 1-D sequence as long as 1024 points would expand to a set over $1.0 \times 10^6$. One method of overcoming memory overflow is to divide the

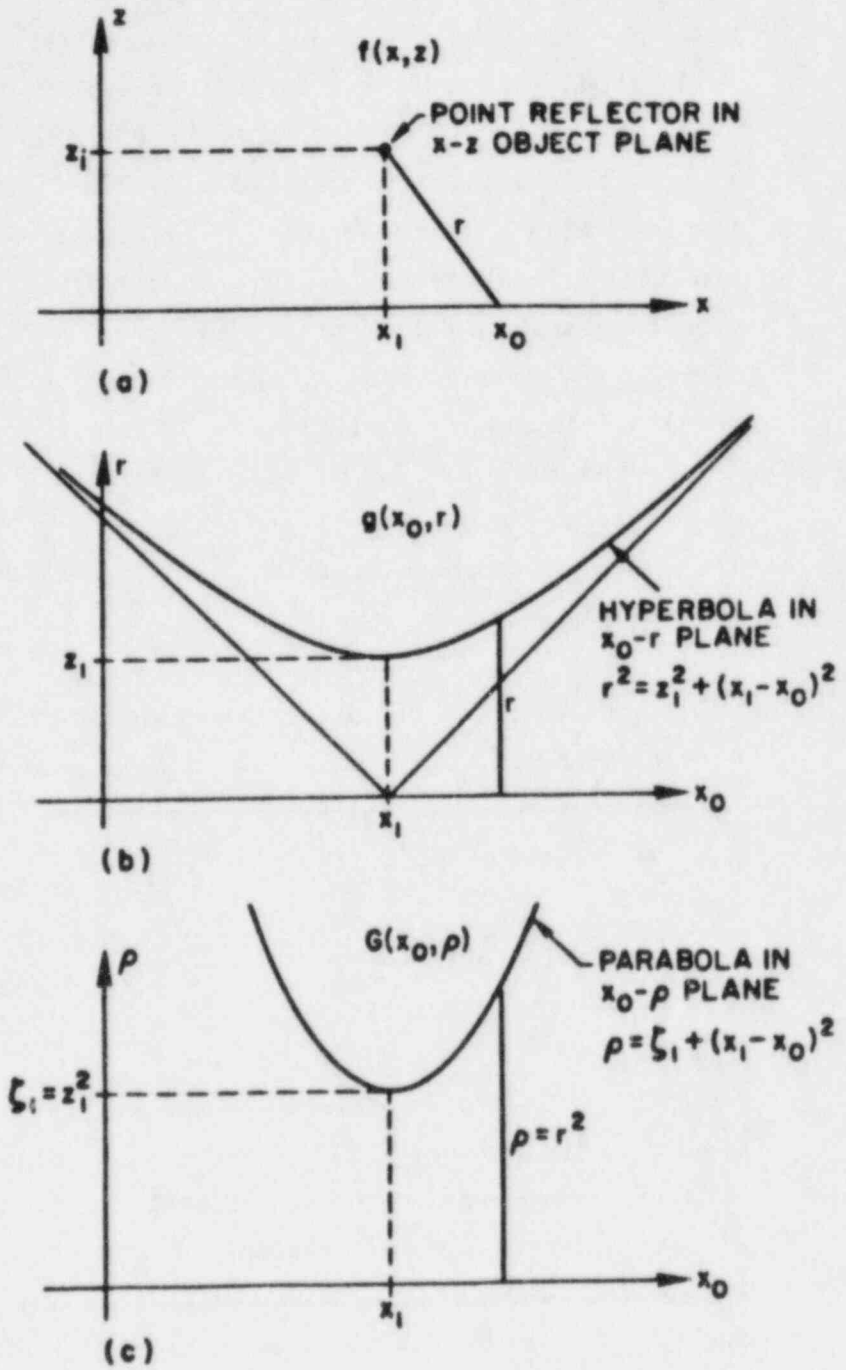



Figure 1. a) Point reflector and recording geometry.
b) Resultant hyperbola generated in $X_0$-r plane.
c) Parabola obtained by plotting the hyperbola versus $r^2$.

volume into planar slices in the Z dimension. In terms of memory space indexing, the scale factor is chosen so that the first two positions in memory, both before and after the transformation are only one index value apart. Specifically, the scale factor is calculated by: scale=1/(2*i+1); where the memory index 'i' represents the first value for each slice in the untransformed space. The transformation essentially distributes the data value to the new memory locations in a fan pattern where the blank spaces are filled in by repeating the data entries. Each slice is then transformed by a different scale factor and an overlap-save FFT type convolution method (Brigham-1974) is used to combine the results. Similarly, overlap-save convolution could be used in the X-Y dimensions as well, but without geometric transformation.

The correlation-reconstruction SAFT processor would execute the following procedures:

1) Fetch slice of data from memory.
2) Geometric transformation in Z by memory mapping.
3) Three-dimensional NTT on data volume slice.
4) Pointwise multiply with transformed PSF and Z-dimension filter.
5) Inverse NTT.
6) Reverse map image slice back into original memory space.

Figure 2 illustrates the organization of the various blocks of memory needed to execute the processing steps listed above. This example represents typical parameters for a pipe inspection situation, where the wall thickness is 2.5 inches, the squint angle is 45 degrees and the reconstruction aperture is 4 inches by 4 inches. The memory block located in the upper left part of the figure represents the unprocessed A-scan data volume. The quantized organization of the data in the memory corresponds to a (40x40x575) element array. For this example, it is assumed that the first 82 X-Y planes (arbitrarally chosen to be 20% of the wall thickness) of

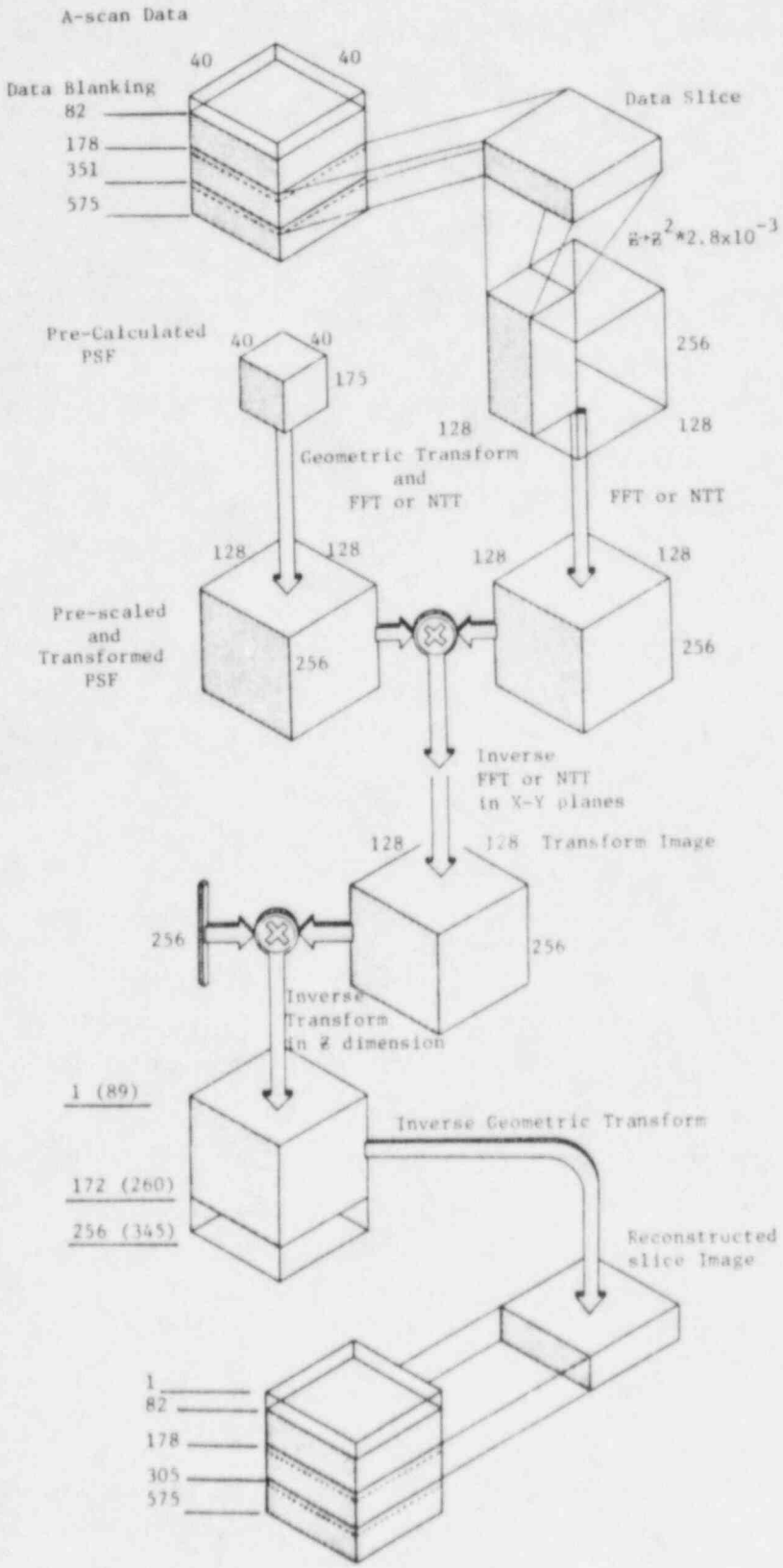



Figure 2. Computational flow diagram for the correlation - reconstruction SAFT algorithm.

the raw data are ignored, because of the potential for data corruption at the specimen surface. The remainder of the volume is divided into three slices, which will be processed sequentially from top to bottom. The number of slices may vary, depending on the particular selection of PSF and NTT parameters, but it is expected that 3-5 slices will probably be used, in most cases.

The figure also shows a (40x40x175) element array containing pre-calculated PSF and the (128x128x256) element slice memory. Each data slice is mapped into the second level memory, as illustrated in the figure, by re-indexing. Two memories are needed, because it will be necessary to save the raw data for the overlap-save process, and because the slice memory represents 12 bit RNS encoded numbers while the raw data is only 8 bits. Next the NTT is executed on each of these memory blocks, followed by pointwise multiplication of the two blocks, which produces a reconstructed slice image (in NTT space). Envelope filtering in the Z-dimension is carried out, by pointwise multiplication, and after inverse transformation of the slice along the X-Y dimensions.

The completely reconstructed and enhanced slice image results after a final Z-dimension inverse transformation and reverse memory mapping (i.e. $Z^2$ geometric transform), which is accomplished during the data transfer to the output image volume. The overlap-save process is indicated by the solid and dashed lines in the last block of the figure. Note that the slice image is returned to the solid line positions, whereas the data was originally taken between the first solid and the second dashed lines.

## 5.0 Computational Architecture

The conceptual hierarchy of computational hardware elements was developed and analyzed during the Phase-I effort. Figure 3 illustrates the total processor architecture, in a simplified form. The imaging system can be broken down into four major parts:

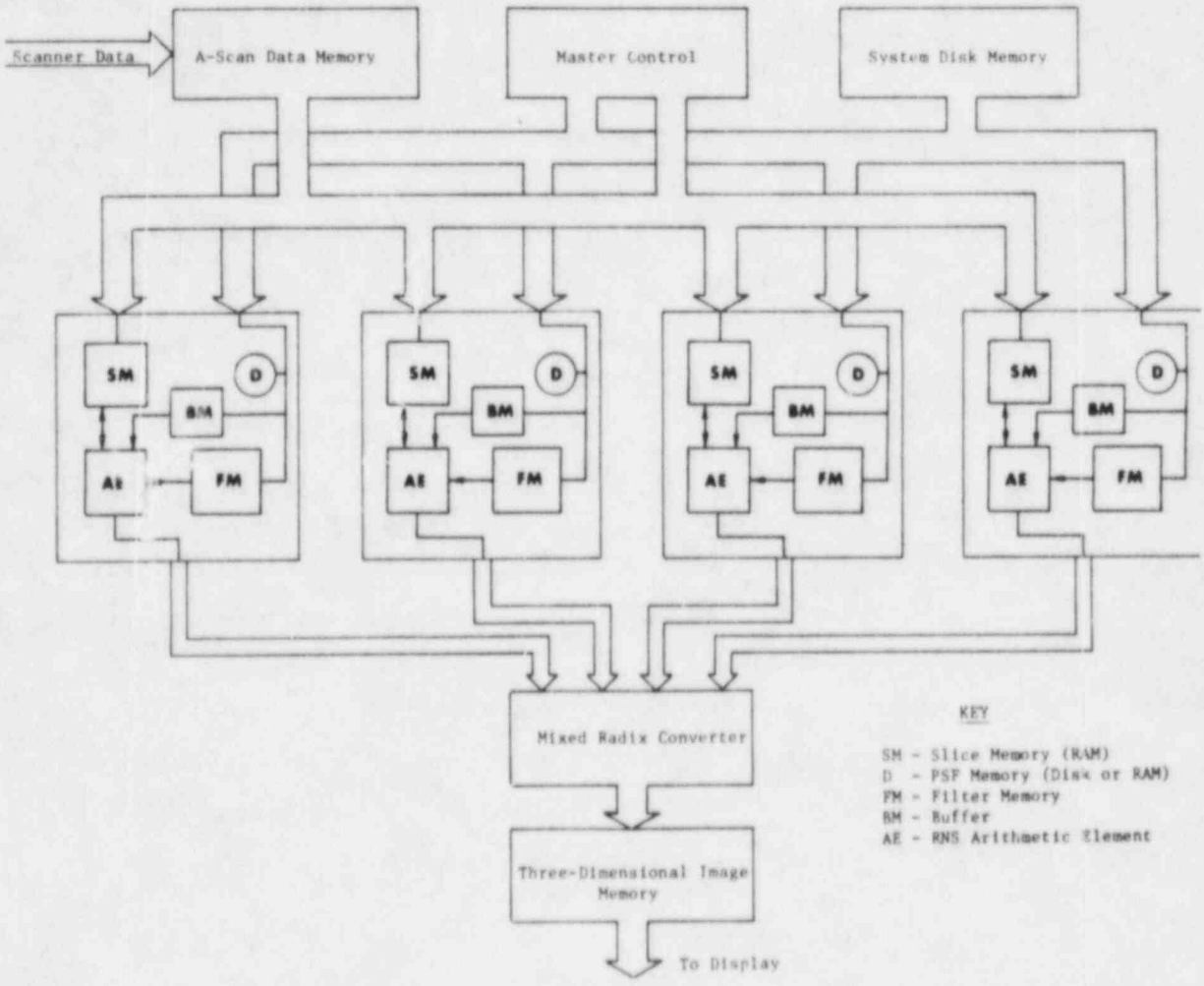



Figure 3. RNS image processor architecture.

input/output, computational memory, and residue number system (RNS) arithmetic element. The intended nature of this processor is to be a self-contained computational system, except for an external data acquisition system (scanner) and possibly a VAX for initialization and disk memory. The figure shows four identical RNS hardware modules with common connections occurring at the A-scan input and the mixed-radix converter output.

The input stage of the system is represented by the scanner for A-scan data collection which generates digital signals for the large RAM memory and analog R.F. signals for magnetic tape archiving. Originally, we thought that the input data might be stored on disk in a digital format, however this appears to be impractical due to the large amount of data to be collected and stored. It should be relatively easy to replay the tape into the processor and reconstruct images either during field inspection or for later analysis. The data acquisition system is expected to transfer 8 bit data serially to the large A-scan memory. The memory will be logically organized as a three dimensional first-in-first-out block of A-scans, arranged in (128x64x1024) format. However, for efficient parallel data transfer to the interior sections of the processor, the memory will be organized into 32 bit words. A disk memory will be needed for system initialization by down-loading point spread function (PSF) values and Z-dimension FIR filter coefficients.

Data output begins at the pipelined mixed-radix converter (MRC) which receives 12 bit data from the four parallel RNS hardware modules. Data within each channel is again packed into 32 bit words for fast transfer to the image memory. This memory is similar in design to the A-scan data memory (i.e. 3-D fifo) and it will occupy the same physical space . Due to the nature of the overlap-save convolution process the input and output memories will be larger than the (40x40) reconstruction aperture, and because the dimensions are rounded up to the closest power of two.

The computational memory is composed of the following: a slice memory (128x128x128x24); a PSF memory, of either 10M byte disk or a RAM of the same size as the slice; a Z-dimension filter memory (128x24); and four fast cache memories (128x24) (Figure 4), within each arithmetic element (AE). The slice memory is organized into four banks in order to increase the throughput by parallel data input, while arithmetic element access is affected through a 4:1 multiplexer interface. The high speed (50ns) cache memories are connected to the RNS arithmetic element in such a way that simultaneous read/write operation and RNS computations can occur. The RNS arithmetic element is also composed of high speed computation devices for performing a number theoretic transform butterfly, a complex multiply, or miscellaneous utility operations.

The overall processor architecture is partitioned into three speed zones: 50ns, 100ns, and 400ns. Each level is controlled by a common bit slice microprocessor, with RNS computations and cache memory exchanges executed during 50ns clock periods, slice memory exchanges executed during 100ns clock periods, and I/O memory exchanges executed during 400ns periods. Data will logically flow through the system by the following sequence:

1. Transfer a slice of data from A-scan to slice memory
2. Transfer 128 (complex) words of x,y or z row data to cache A (see Fig.4)
3. RNS element reads data pairs from A and writes to C
4. Transfer next row of data from slice to cache B while RNS AE operates on cache A to C
5. RNS AE operates on cache B to D while results are transferred from cache C to slice memory, and new data is transferred from slice memory to cache A
6. The cycle is repeated until all transformations or multiplications are complete.

## 6.0 RNS Arithmetic Unit

The arithmetic unit (Figure 4), within the processor's high speed module (20MHz), was divided into three types: basic arithmetic elements, an NTT butterfly element, and cache memories. From an understanding of these elements it was possible to estimate the overall performance of the complete processor and to determinewhether it has the potential for enhancing the performance of a real-time inspection system.

### RNS arithmetic elements

The basic RNS arithmetic elements are the foundation of the computational process, so these were analyzed first by considering the processor's needs and the latest available research results for these devices.

For a modulus $m<2^6$, the complete tables for an arithmetic operation can be stored in a commercially available high speed PROM of size (4096x8) bits or smaller. These components can be clocked at 20 MHz rates, permitting extremely fast arithmetic at reasonable cost. Sufficiently high-speed integrated memories are not available in sizes to support larger moduli. Designs of large moduli RNS computational elements have recently appeared in the literature by four authors. Jenkins' approach (Jenkins-1982) has been selected as the preferred approach due to its simplicity and speed. Jenkins' standard computational element (SCE) is shown in Fig. 5 for a modulus which can be represented with b-bits. The adder is a b-bit 2's complement adder. By storing different lookup tables in the PROM, the computational element can be configured as a modulo adder, subtractor, or mixed radix conversion kernel. If the modulus is prime, the SCE can be used for general multiplication. Complex arithmetic can be performed by using SCE's to compute the real and imaginary parts of a computational result separately.

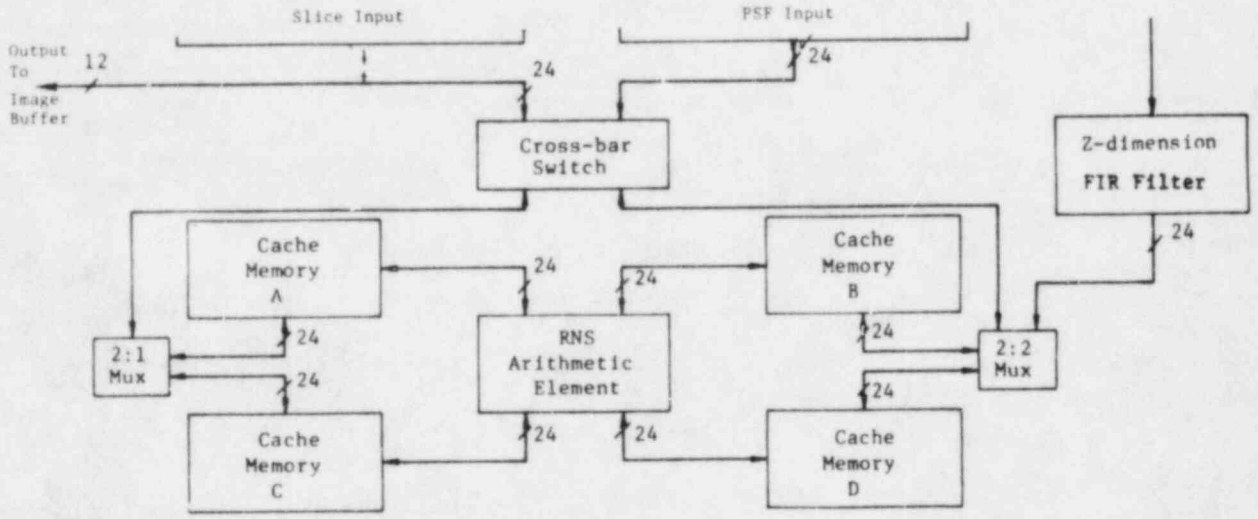
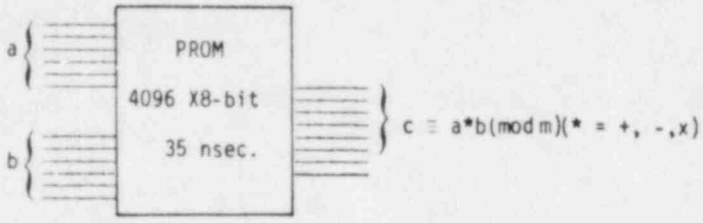



Figure 4. High speed arithmetic unit for the RNS processor (Fig. 3).

Small Moduli ($m \leq 64$)




Large Moduli ($m \leq 1000$)

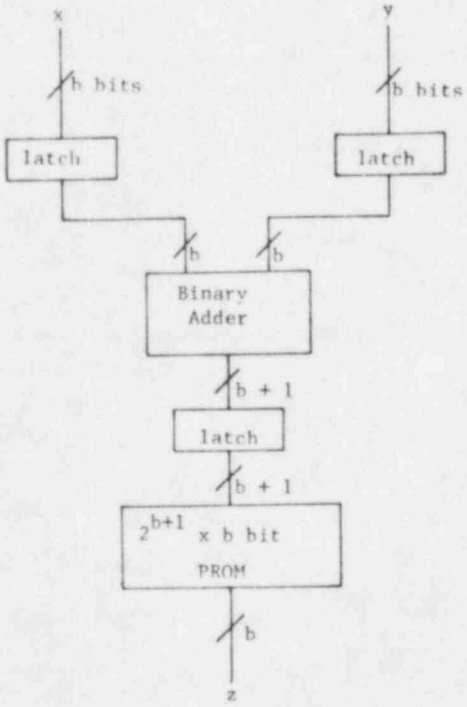
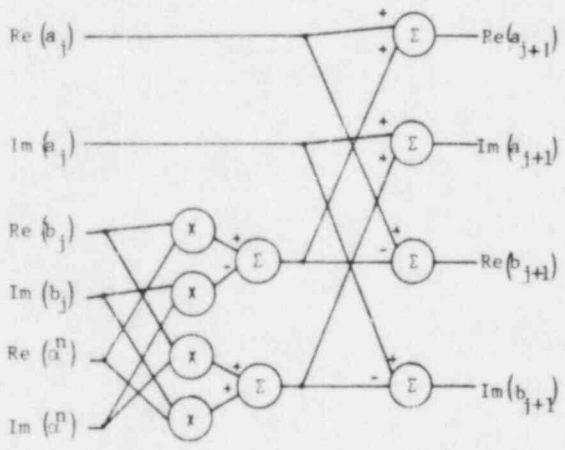



Figure 5. Hardware Implementations of RNS Operations.




$$a_{j+1} = a_j + b_j\alpha^n$$

$$b_{j+1} = a_j - b_j\alpha^n$$

Figure 6. Complex NTT Butterfly Element.

We next show how the PROM can be programmed to create an adder from the SCE. The two inputs x and y are added by the binary adder to give the intermediate result u. The PROM then performs the following mapping by a table lookup

$$u \longrightarrow \begin{cases} u & \text{if } u < m \\ u - m & \text{if } u > m. \end{cases}$$

Multiplication is performed by using index calculus (finite field logarithms). In this method, multiplication modulo p is transformed to addition modulo (p-1). The two inputs x and y are converted to the index domain using the PROM tables from the previous arithmetic operation. These indices are then added modulo (p-1) and the PROM performs the inverse index lookup. For moduli less than 1024, b = 10 bits are required. The adders would be built from three 4-bit adders and the PROM would be constructed from two PROM's, one of size (2048x8) and the other of size (2048x4). The latch/adder and latch/PROM operations can each be performed in 50 nano-seconds using these components. Throughput rates of 20 MHz may be achieved by pipelining the SCE with a 2-cycle latency for single operations.

**NTT butterfly unit**

The SCE's are used to construct an NTT butterfly in the RNS arithmetic unit in such a way as to produce basic transform operations in an efficient manner. The complex NTT butterfly element is illustrated in Fig. 6, where each operation shown is an appropriately programmed SCE. Assuming a 50 nano-second clock cycle, the SCE's will take 2 clock cycles to execute an operation, and the full butterfly will require 6 cycles. The butterfly arithmetic element is fully pipelined, which means that it will begin a new butterfly every clock cycle for a 20 MHz data throughput. The number of SCE's required is 10, for a total number

of integrated circuits given by

| | |
|---|---|
| 4-bit 2's complement adders | 30 |
| 2048 x 8 bit PROM's | 10 |
| 2048 x 4 bit PROM's | 10 |
| Latches | 30 |
| Total IC's | 80 |

## 7.0 Processor Performance

Based on the assumed computational rates of the RNS arithmetic elements, we were able to estimate the total processor performance, in terms of A-scans per second. This figure of merit is used so that a relatively unbiased performance characteristic can be generated, and which is not dependant on the X-Y scanning density or the scanner speed. The two critical areas of the computational process, which will have the greatest effect on the image output rate will be memory transfers and RNS arithmetic operations.

The time dominant RNS operation will be the NTT. A 128 point transform has seven computational stages with each stage requiring 3.45 micro-seconds for a total time of 24.15 micro-seconds. These timings assume a basic clock period of 50 nano-seconds. We next use this transform time to estimate the time it will take to process a complete data slice. First, the X-Y dimensional planes are transformed in a time proportional to (128x2x150), which yields 927 milli-seconds. The dimension 150, is a typical worst case Z-dimension of the A-scan data slice before $Z^2$ geometric tranformation by memory index mapping. The last step is to NTT in the Z-dimension, which is proportional to (128x128), and requires 397 milli-seconds, giving us a total of about 1.32 seconds for the entire slice transform. It has been determined that the PSF can be pre-transformed in the X-Y dimension before down loading into the processor memory during system initialization. Therefore, its Z-dimension NTT will take another 397 milli-seconds. The

correlation process for image reconstruction corresponds to a simple pointwise multiplication (in NTT space) and will be proportional to (128x128x128), which for 50 nano-second time periods gives 0.1 seconds. A one dimensional Z-dimension envelope filtering operation will take another 0.1 seconds. Finally, the slice must be inverse transformed in X-Y-Z, and this will require an additional 1.32 seconds. These various timings are applied to each slice of data which pass through the RNS computational unit, and it is expected that five slices will typically be needed to completely process the A-scan data volume. Therefore, it will take about 16.2 seconds for an image to be produced. These results are summarized in the following table:

**Arithmetic Operations**

| | |
|---|---|
| Single 128 point complex NTT | 24 micro-seconds |
| Slice transform | |
| (x-y, 128x2x150) | 0.93 |
| ( z, 128x128) | 0.40 |
| PSF transform | |
| (x-y, pre-transformed) | 0.0 |
| (z, 128x128) | 0.40 |
| Pointwise multiply slice by PSF | |
| (128x128x128) | 0.10 |
| Z-dimension envelope filter | 0.10 |
| Inverse slice NTT | 1.32 |
| Total | 3.24 |
| For five slices per A-scan volume | |
| (5*3.24) | **16.2** seconds |

Memory transfers of data into and out of the processor will add a very small overhead to the arithmetic processing time. Typically 77 milli-seconds will be needed to transfer each (128x40x150) element slice into the RNS memory modules from A-scan input buffer;

similarly, each output slice image will take slightly less time. Since an overlap-save process will be used to produce continuous images in the X-Y dimension, a smaller number of image pixels will be sent to the image buffer than raw data was transfered in. At output, only a (40x89) X-Y image will be used at any one time. Data transfers from the slice memory to the high speed cache will be time shared with the NTT and pointwise multiplication, so these are not counted. Similarly, transfers from the PSF to the cache will also be time shared. The following table list the relative timings for the various memory processes without considering any time sharing optimization. A alternate disk transfer method is also included for comparison.

**Memory Transfers**

A-scan to slice
(128x40x150) ---------- 0.077
Slice to image buffer ---------- 0.077
PSF to cache with RAM
(without time sharing)
(128x128) ---------- 1.60
Total **1.75** seconds

PSF to cache with disk
To buffer (128x128x16ms/21) --- 12.5
Buffer to cache (128x128x100ns) 1.6
Total **14.1** seconds

During the course of this design exercise, in Phase-I, it was determined that the PSF memory might not have to be made up of 64K RAM chips, but that disk memory could offer a lower cost (and possibly lower risk) option for a prototype system. The PSF memories (one for each RNS module) are needed only for reading data, not reading and writing like the slice memory. A considerable savings could be realized by using small 10 Mbyte Winchester disk drives in place of the RAM chips. In addition to cost, much less

space would be needed to house the disk memory, they would use less power, and they would require less labor during processor fabrication. Their nonvolatile storage capability might be useful during repeated imaging with a fixed PSF. However, the penalty for using these devices would be that the processor performance would be degraded. Each set of (128x128) data transfers would take about 14 seconds, and five of these transfers would total over 70 seconds. This includes buffering the transfer through a 21 word intermediate memory, before going into the high speed cache. This would allow 21 transfers per disk revolution (16 milli-seconds). After evaluation of the prototype is completed, and if it is determined that a faster system would provide a significant benefit, then the disk drives could be replaced with RAM. The impact on the processor architecture would be minimal, and the only added burden would be extra cost for the RAM chip plus labor. One might expect that by the time a decision is made to upgrade the system design, new generation 256K RAM chips might be commonly available and at a lower price than today.

The overall system performance can therefore be expressed as about 24 seconds per image with RAM and 87 seconds with disk. These times translate into 145 A-scans per second and 40 A-scans per second, respectively; and would represent the continuous image production rates for typical pipe inspection cases of 0.6, 0.75, 1.5, and 2.5 inch wall thickness. Similar rates are expected for typical (12x12x12) inch pressure vessel volumes.

The following table compares these conclusions with other processing methods:

**Comparison of Various Throughput Rates**

| | A-scans per second |
|---|---|
| VAX 11/780 | 0.4 |
| Coherent-sum SAFT | |
| Current 4 processor model | 10 |
| Advanced 16 processor model | 40 |
| Correlation-reconstruction RNS SAFT | |
| With disk drives for the PSF | 40 |
| Advanced RAM model | 145 |

An equivalent inspection rate for the advanced RNS SAFT processor, applied to a 12"x12"x12" pressure vessel volume is:

Total volume -------- 1728 cubic inches

x-y sampling density --0.1 inches

$$\frac{60 \times 145 \times 1728}{120 \times 120} = \mathbf{1044} \text{ cubic inches per minute}$$

## 8.0 References


E. O. Brigham, **The Fast Fourier Transform**, Prentice-Hall, Inc., New York, p. 212 (1974).

A. Z. Baraniecka and G. A. Jullien, "Residue Number System Implementation of Number Theoretic Transforms in Complex Residue Rings", IEEE Trans. ASSP, 28,285(1980).

W. K. Jenkins, "A Standard Computational Element for the VLSI Realization of Digital Processors Using Modular Arithmetic," Sixteenth Asilomar Conference on Circuits, Systems, and Computers, Pacific Grove, CA (Nov. 1982).

G. A. Jullien and W. C. Miller, "An RNS Arithmetic Unit for 2-D Digital Convolution," Preprint (1982).

J. V. Krogmeier and W. K. Jenkins, "Error Detection and Correction in Quadratic Residue Number Systems," 26th Midwest Symposium on Circuits and Systems, Puebla, Mexico (August 1983).

S. H. Leung, "Application of Residue Number Systems to Complex Digital Filters," Proc. 15th Asilomar Conf. Circuits, Systems, and Comput., Pacific Grove, CA., 70-74 (Nov. 1981).

S. J. Norton, **Theory of Acoustic Imaging**, Stanford Electronics Laboratories, Report No. 4956-2 (1976).

N. S. Szabo and R. I. Tanaka, **Residue Arithmetic and its Application to Computer Technology**, McGraw-Hill, 1967.

F. J. Taylor and C. H. Huang, "A Floating Point Residue Arithmetic Unit," J. Franklin Institute, 311, 33 (1981).

B. D. Tseng, G. A. Jullien, W. C. Miller, "Implementation of FFT Structures Using the Residue Number System, IEEE Trans. Comput., C-28, 831 (1979).

## APPENDIX A

Given a fixed integer m>1 (the **modulus**) and any integer a, the Euclidian algorithm asserts the existence of integers q (the quotient) and r (the remainder) such that

$$a = q \cdot m + r \ , \quad \text{where } 0 \leq r < m.$$

The integer r is called the **residue** of a modulo m. If two integers a and b have the same residue modulo m, they are called congruent modulo m or are said to belong to the same **residue class** modulo m, denoted $a \equiv b \pmod{m}$. For an integer a, $|a|_m$ denotes the set of all integers belonging to the same residue class as a, and a or any other integer in the class is called a representative of $|a|_m$. Arithmetic operations of residue classes can be easily defined to form a residue class ring denoted $Z_m$:

$$Z_m = \{ \, |r|_m \mid r = 0, 1, \ldots, m-1 \, \}$$

with operations defined by

$$|a|_m + |b|_m = |a + b|_m$$
$$|a|_m - |b|_m = |a - b|_m$$
$$|a|_m \cdot |b|_m = |a \cdot b|_m,$$

i.e., the sum, difference, or product of two residue classes is the residue class of a sum, difference, or product of their representatives. For an integer a which is relatively prime to m, there exists an integer b for which $|ab|_m = 1$, i.e., a has a multiplicative inverse b, usually denoted $|1/a|_m$. However, $|b|_m |1/a|_m = |b/a|_m$ if and only if a divides b with no remainder.

A residue number system is constructed by choosing relatively prime moduli $m_1, \ldots, m_k$ and forming the product of the residue rings

$$Z_{m1} \times Z_{m2} \times \ldots \times Z_{mk} = \{ \, (r_1, r_2, \ldots r_k) \mid r_i \in Z_{mi} \, \}$$

Chinese Remainder Theorem. Let $m_1,\ldots,m_k$ be relatively prime, positive integers. Then for any k remainders $r_1,\ldots,r_k$ with $0 \le r_i < m_i$, there exists a unique integer

$$x \in \left[0 \ , \ \prod_{i=1}^{k} m_i - 1\right]$$

for which

$$x = r_i \pmod{m_i} \text{ for } i=1,\ldots,k.$$

Consider a sequence of arithmetic calculations on a set of integer inputs $I_1,\ldots,I_t$. For simplicity, assume all inputs and the output are non-negative and smaller than the product of all the moduli. One way to proceed is to perform the arithmetic directly. The other way is to encode each $I_j$ into RNS

$$I_j \quad (I_j \pmod{m_1},\ldots,I_j \pmod{m_k}),$$

perform the arithmetic componentwise without carries, obtaining the residue encoded output. The Chinese Remainder Theorem tells us that this output can be uniquely decoded, yielding the identical answer as the direct calculation.

EXAMPLE. Evaluate the expression $(31 \cdot 4 + 85)/11$ using the RNS with $m_1 = 3$, $m_2 = 5$, and $m_3 = 7$.

SOLUTION. The problem would not be legitimate if either 11 were not relatively prime to each modulus or if the final answer were not an integer (a quick check shows that it is). In practice, general divisions are avoided. Note also that intermediate calculations overflow the range [0,104] of the system, but since the final answer (viz.,19) is within the range, this is permissible. We begin by encoding:

$$31 \text{ ----> } (1,1,3)$$
$$4 \text{ ----> } (1,4,4)$$
$$85 \text{ ----> } (1,0,1)$$
$$|1/11| \text{ ----> } (2,1,2)$$

Encoding the multiplicative inverse of 11 is more obscure than the others, but it's quite easy to check the encoded form. Since $11 \leftrightarrow (2,1,4)$, $11 * |1/11| \leftrightarrow (2,1,4)*(2,1,2)=(1,1,1)$. The computation becomes

$$(1,1,3) \text{ <---- } 31$$
$$\underline{(1,4,4)} \text{ <---- } 4$$
$$(1,4,5)$$
$$\underline{(1,0,1)} \text{ <---- } 85$$
$$(2,4,6)$$
$$\underline{(2,1,2)} \text{ <---- } |1/11|$$
$$(1,4,5) \text{ ----> } ?$$

The final step is to decode (1,4,5). There are two algorithms for doing this -- one is based on the proof of the Chinese Remainder Theorem and the other converts from the RNS to a mixed radix form (see (Szabo, 1967) for a complete discussion of both methods). Regardless of the algorithm used, communication between the different residue class rings is required. This communication can be considered as conceptually analogous to carrying in normal radix calculations (where the communication is between the columns). However, in the residue system this communication is delayed. Although this may be inefficient for performing only a single arithmetic operation, the total system efficiency significantly improves as longer strings of calculations are performed before an output is required.

For the mixed radix conversion, we will find integer coefficients $a_1$, $a_2$, and $a_3$, $0 \leq a_i < m_i$ for which the residue triple (1,4,5) can be expressed as

$$X = a_1*5*7 + a_2*7 + a_3.$$

The mixed radix weights 5*7, 7, and 1 are derived from the moduli set. Notice that $x=a_3 \pmod 7$, so $a_3 = 5$. Continuing,

| Operation | Algebraic Form | Residue Modulo 3 | 5 | 7 | Solution |
|---|---|---|---|---|---|
| | $X = a_1*5*7 + a_2*7 + a_3$ | 1 | 4 | 5 | $a_3 = 5$ |
| subt. $a_3$ | | - 2 | 0 | 5 | |
| | $X-a_3 = a_1*5*7 + a_2*7$ | 2 | 4 | | |
| mult. 1/7 | | X 1 | 3 | | |
| | $(X-a_3)/7 = a_1*5 + a_2$ | 2 | 2 | | $a_2 = 2$ |
| subt. $a_2$ | | - 2 | 2 | | |
| | $(X-a_3)/7-a_2 = a_1*5$ | 0 | | | |
| mult. 1/5 | | X 2 | | | |
| | $((X-a_3)/7-a_2)/5 = a_1$ | 0 | | | |

Hence, (1,4,5) $X = 0*5*7 + 2*7 + 5 = 19$, is the correct answer.

To use a conventional RNS for complex arithmetic, rings $Z_{m_i}[j]$ are defined by

$$\{ x + jy \mid x,y \in Z_{m_i} \}$$

with $j_2 = -1$. For integers a and b, $0 \leq a,b < M$ with

$$a = (a_1, \ldots, a_k),$$
$$b = (b_1, \ldots, b_k),$$

and

$$a_i = |a|_{mi},$$

$$b_i = |b|_{mi},$$

the complex number a + jb is encoded as

$$(a + jb)_i = a_i + jb_i$$

with operations defined componentwise by

$$(a_i + jb_i) + (c_i + jd_i) = (a_i + c_i) + j(b_i + d_i)$$

$$(a_i + jb_i)(c_i + jd_i) = (a_ic_i - b_id_i) + j(a_id_i + b_ic_i)$$

with the sums and products on the right hand side occurring modulo $m_i$.

## APPENDIX B

### Number Theoretic Transform

The NTT is an analog of the discrete Fourier transform with the complex field replaced by a finite field. A field is an algebraic system over which two operations, usually denoted addition and multiplication, are defined. A number of conditions on these operations are required: the existence of the additive and multiplicative identities 0 and 1, commutitive and distributive properties, and the existence of additive and multiplicative inverses. The real numbers and the complex numbers are examples of fields, but the integers are not since the multiplicative inverse of an integer is not, in general, an integer.

For any prime p, the ring $Z_p$ of integers modulo p is a field containing p elements. Extension fields of $Z_p$ can be formed, and the number of elements in such a field is $p^n$ for a positive integer n. All finite fields are of this form, and are usually denoted $GF(p^n)$ (the Galois field of $p^n$ elements).

As an elementary example, we will construct the fields GF(9). The base field is taken as $Z_3 = \{0, 1, 2\}$, with addition and multiplication defined modulo 3. We observe that the polynomial $x^2+1$ has no roots in $Z_3$. Let $Z_3[x]$ be the ring of all polynomials in the indeterminant x with coefficients from $Z_3$. A new ring now is formed by performing arithmetic in $Z_3[x]$ modulo the polynomial $x^2+1$. This new ring (actually a field) is denoted $Z_3[x]/(x^2+1)$. Any element of this field can be represented as ax + b, for a, b $Z_3$.

The sum of two elements ax + b and a´x + b´ is (a + a´)x + (b + b´), where the coefficients are added modulo 3. Two elements

are multiplied as polynomials, and the product is taken to be the remainder upon dividing this polynomial by $x^2+1$, or equivalently, by setting $x^2 = -1$ and combining terms. For example,

$$\begin{aligned}(2x + 1)(x + 2) &= 2x^2 + 5x + 2 \\ &= 2x^2 + 2x + 2 \\ &= 2 (-1) + 2x + 2 \\ &= 2x .\end{aligned}$$

Every nonzero element has an inverse; for example,

$$(2x + 1)^{-1} = 2x + 2 \text{ since } (2x + 1)(2x + 2) = 1.$$

While the NTT is used to perform circular convolutions of sequences of ordinary integers, all intermediate arithmetic in the computational process is performed over a finite field. The definition of the NTT is given next. An element of a finite field F is said to have order N if $^{N} = 1$ and $^{k} \neq 1$ for $1 < k <$ N. If we have a sequence x(n) of integers of length N the NTT of x(n) is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) \quad ^{nk}$$

with inverse

$$X(n) = N^{-1} \sum_{k=0}^{N-1} X(k) \quad ^{-nk}$$

The transformed sequence X(k) does not have a physical interpretation similar to the Fourier transform; however, the NTT does satisfy the cyclic convolution property, viz.,

$$NTT(h * x) = NTT(h) \text{ X } NTT(x).$$

Hence, the convolution of two sequences is obtained as the inverse

NTT of the pointwise product of the NTT's of the two sequences. Moreover, if N (the order of) is highly composite (preferably a power of 2), then the NTT can be efficiently computed in a manner analogous to the FFT. There are two principle advantages to the use of the NTT over the FFT for performing convolutions. First, the arithmetic is exact since it is performed over finite fields. By contrast, using fixed point arithmetic for the FFT, rounding or truncation occurs after each multiplication and in the approximation of the twiddle factors. This could pose a serious problem using a short word length FFT repeatedly over a large three-dimensional data set. Secondly, an RNS implementation of the FFT would require extensive use of scaling. The scaling operation is cumbersome and time consuming in an RNS, requiring in effect, a decoding, scaling, and re-encoding procedure. A residue implementation of the NTT requires no scaling. The dynamic range of the RNS is selected to be large enough to contain the result of the final convolution; intermediate overflows in the NTT, multiply-inverse, NTT process are of no consequence.

Use of RNS methods to compute complex convolutions has been discussed extensively in (Baraniecka-1980). It is shown that to compute a complex NTT-convolution of length N, the moduli $p_i$ must be prime and chosen as follows.

* If $p_i = 4n + 1$, then N must divide $p - 1$.

* If $p_i = 4n + 3$, then N must divide $p^2 - 1$.

The terms in the final convolution must be contained in a range spanned by the product of all moduli. Primes less than 1000 which admit an NTT of length 256 (or any factor of 256) are 257 and 769 (both of the form 4n+1) and 127 and 383 (both of the form 4n+3). The product of these four moduli exceeds 33 bits. The next section demonstrates that this dynamic range is sufficient.

**Dynamic Range Requirements for Correlation Reconstruction with Envelope Detection**

An x-y data aperture of (40x40) is assumed for pipe inspection situations with a dynamic range for the raw data of 8 bits. Dynamic range requirements can be estimated using either the coherent summation or correlation reconstruction algorithms since the final results will be the same. The coherent summation approach simplifies the estimate. For reconstruction by coherernt summation, (40x40)=1600 data values are added together to form each pixel of the reconstructed image. The worst case dynmaic range requirement would, therefore, be $\log_2(8x1600)=18.6$ bits. If the PSF were chosen to consist of only 0's and 1's (which is all that is needed to duplicate coherent summation by correlation reconstruction), 18.6 bits would suffice for correlation reconstruction (without envelope detection). If 4 bits are assumed for the PSF representation (improving the quality of the reconstructed image), then 22.6 bits are required. Finally, for envelope detection, a FIR space filter will be applied to the data. It has been demonstrated by Dr. Busse at Battelle Northwest that a complex filter with a 16-point impulse response will suffice. If 8 bits are assumed for the filter coefficients, then in the worst case the dynamic range would grow by 9 bits ($=\log_2[16x8x2x2]$), with the two factors of 2 accounting for the real additions required for a complex multiply and the addition of the magnitudes of the real and imaginary parts of the filtered result. Hence, the worst case dynamic range requirement is 31.6 bits. In practice, the extremes will never be attained, and it will be possible to increase the accuracy of the PSF coefficients or the filter coefficients.

NRC FORM 335 (11-81) U.S. NUCLEAR REGULATORY COMMISSION

# BIBLIOGRAPHIC DATA SHEET

1 REPORT NUMBER (Assigned by TIDC and Vol. No. (if any))
NUREG/CR-4170

2 Leave blank

3 TITLE AND SUBTITLE
An Ultra-High Speed Residue Processor for SAFT Inspection System Image Enhancement

4 RECIPIENT'S ACCESSION NUMBER

5 DATE REPORT COMPLETED
MONTH: May YEAR: 1984

6 AUTHOR(S)
J. N. Polky
D. D. Miller

7 DATE REPORT ISSUED
MONTH: March YEAR: 1985

8 PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Sigma Research, Inc.
Seattle Laboratory
565 Industry Drive
Seattle, Washington 98188

9 PROJECT/TASK/WORK UNIT NUMBER

10 FIN NUMBER
D1133

11 SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Division of Engineering
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

12a TYPE OF REPORT
Technical

12b PERIOD COVERED (Inclusive dates)
October 1983 to March 1984

13 SUPPLEMENTARY NOTES

14 ABSTRACT (200 words or less)

The Phase-I feasibility study of residue number system (RNS) image processing for SAFT inspection has successfully determined that an advanced inspection system may be built using a correlation-reconstruction SAFT algorithm, implemented with RNS techniques and off-the-shelf electronic components. Images are reconstructed in a number theoretic transform domain with simple pointwise multiplication of the A-scan data volume by a custom point spread function (PSF), all in a highly parallel computational architecture. These methods also allow image enchancement to be easily performed for improved flaw visualization, and with negligible speed reduction. It has been determined that high resolution three dimensional flaw images may be generated and that a commercially viable product could result through development of a prototype real-time RNS processor. The hardware is expected to be made up of 100 nsec bit slice microprocessor components and large RAM storage units. Based on the performance estimates of the Phase-I effort, this new image processing system has the potential to acquire and focus the equivalent of the 145 A-scans per second, which translates into more than 1000 cubic inches per min. inspection rate for typical pressure vessel specimens.

15a KEY WORDS AND DOCUMENT ANALYSIS

Ultrasonic imaging
Ultrasonic testing
Residue number system
Signal processing
SAFT-UT

15b DESCRIPTORS

16 AVAILABILITY STATEMENT
Unlimited

17 SECURITY CLASSIFICATION (This report)
Unclassified

18 NUMBER OF PAGES

19 SECURITY CLASSIFICATION (This page)
Unclassified

20 PRICE
$

*U.S. GOVERNMENT PRINTING OFFICE: 1985-461-721:20013

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555