

1990

Investigation of techniques for high speed CMOS arbitrary waveform generation

Albert Henry Nehl
Portland State University

Let us know how access to this document benefits you.

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

 Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Nehl, Albert Henry, "Investigation of techniques for high speed CMOS arbitrary waveform generation" (1990). *Dissertations and Theses*. Paper 4109.

10.15760/etd.5993

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

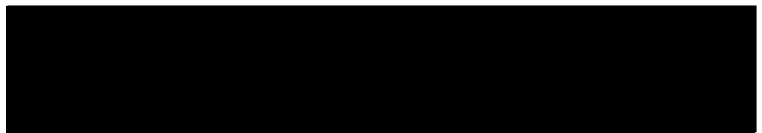
AN ABSTRACT OF THE THESIS OF Albert Henry Nehl for the Master of Science in Electrical and Computer Engineering presented February 2, 1990.

Title: Investigation of Techniques for High Speed CMOS Arbitrary Waveform Generation

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:



W. Robert Daasch, Chair



Rolf Schauman



Faryar Etesami

Today a growing number of applications in design engineering, production and environmental testing, and system service require specific analog waveforms and digital patterns. Such requirements are neither satisfactorily nor easily met by the use of standard function or single purpose, custom generators.

Traditional methods of waveform generation suffer from undesirable complexity or mediocre performance and are otherwise limited. For the majority of arbitrary waveform generation applications, including medical engineering, modal analysis and electronic engineering, direct digital synthesis techniques are satisfactory. Direct digital synthesis, based generally on periodic retrieval of predetermined amplitude values, may be used to generate such waveforms. Within the limits imposed by the system's maximum sample rate and the Nyquist criteria, any waveform may be produced using these techniques.

The objective of this inquiry, within a particular set of constraints, is to extend the cost/performance envelope of direct digital synthesis techniques for the generation of arbitrary waveforms. Performance is enhanced, particularly in the areas of output bandwidth and signal purity.

A single ASICs will implement all DAWG functionality, except waveform datapoint memory, digital to analog conversion, and post-conversion filtering, using an industry standard process. Access to a useful set of waveforms, including those of complex symmetry, indicates the use of memory for their storage. The system provides features and performance such as standard functions: sine, cosine, rectangular, triangular, and sawtooth waves; arbitrary, user defined or captured, waves; extensive memory capacity; random access to waveform segments; datapoint segment sequencing and looping; free running, gated, triggered or swept modes; greater than 60 dB signal

to noise ratio, or SNR; and an output frequency, for simple, highly symmetric waveforms such as sine, in excess of thirty megahertz.

Using a words-wide memory configuration, datapoint word groups may be recalled and individual datapoint words may be multiplexed or shifted to the digital to analog converter at some appropriate sample rate.

An algorithm which determines an optimal number of samples per cycle and a sample clock rate, while minimizing the difference between the frequency produced and the original target frequency is developed. The number of samples per cycle and the sample clock rate are both functions of the system's maximum number of points per cycle, the sample clock range and its granularity, as well as the target frequency.

These improvements provide for improved accuracy, long, possibly aperiodic, waveforms, and an extended output bandwidth. A new ring oscillator delay element and associated bias circuitry are developed. The result of combining these several developments in a novel structure is the DAWG, or Digital Arbitrary Waveform Generator. This system, although conceptually simple, is capable of performance which matches or exceeds that of many currently available AWGs.

INVESTIGATION OF TECHNIQUES FOR HIGH SPEED CMOS
ARBITRARY WAVEFORM GENERATION

by

ALBERT HENRY NEHL

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
IN
ELECTRICAL AND COMPUTER ENGINEERING

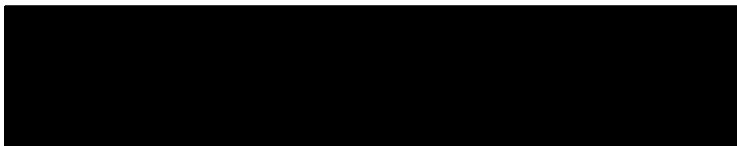
Portland State University
1990

TO THE OFFICE OF GRADUATE STUDIES:

The members of the committee approve the thesis of Albert Henry Nehl presented February 2, 1990.



W. Robert Daasch, Chair

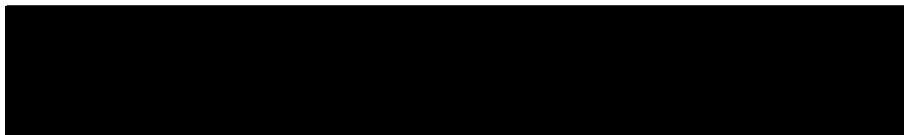


Rolf Schauman



Faryar Etesami

APPROVED:



Rolf Schauman, Chair, Department of Electrical Engineering



C. Willam Savery, Interim Vice Provost for Graduate Studies and Research

DEDICATION

This thesis is dedicated to my mother, Rita Nehl, who, by faith and example, has given immeasurably to my love and understanding of life.

This thesis is also dedicated to the spirit of love and compassion upon which the soul of mankind depends. If the substance of this work seems far removed from that spirit, be not misled. Truth, in time, delivers, with each creative act, one more step towards enlightenment.

ACKNOWLEDGMENTS

It is with pleasure that I acknowledge and thank the many others without whose help and encouragement this work would have been more burdensome and less satisfying.

Dr. W. Robert Daasch, who has served as my graduate advisor for the last two years, has provided many helpful suggestions and served as a sounding board for many of my ideas. This process of stimulation and reflection has been instrumental in the resolution of several key issues during the design process.

It is with pleasure, respect, and admiration that I thank the members of Tektronix's Advance Development Group, ADG, who have not only selflessly assisted me in my research and design efforts, but have further assisted, by way of review and comment, in the preparation of this thesis. In particular I wish to thank Vince Ast, Dave McKinney, Tony Rick, Tim Sauerwein, and Chuck Saxe. Without their support this effort would not have been possible. In addition to the members of ADG there are two individuals within Tektronix whose help has been invaluable. Fred Azinger has been a ready and able collaborator whose suggestions have contributed to problem analysis, definition, and numerous corrections and clarifications of this text. Skip Hillman's review of the draft thesis and subsequent suggestions helped me to correct or clarify several points including an important basis case for the analysis of the carry-delay adder.

Among the students with whom I have worked during the course of my studies at Portland State University I have enjoyed the assistance and support of many. In particular I would like to thank Bret Leichner, Linda Schaefer, David Smith, and Brad Thomas for their patience, interest, and suggestions.

I would also like to thank three faculty members, from diverse disciplines, Diana Burn, Don Moor, and Susan Karant-Nunn. These three, from the departments of mathematics, philosophy, and history, have been particularly important to my success at this institution, and have engendered in me an understanding of the university as a social institution and as a personal and communal tool for the reformation of both the individual and for society.

Finally, I would like to thank all those within the university and the community who have made this experience possible but whose names have here gone unmentioned.

Thank you all for everything you have done. To the extent of my abilities I will pass your favors, in kind, to those who may find them of use.

TABLE OF CONTENTS

	PAGE
DEDICATION.....	iii
ACKNOWLEDGEMENTS.....	iv
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
CHAPTER	
I INTRODUCTION.....	1
II PROBLEM DEFINITION.....	4
III SURVEY OF SELECTED LITERATURE.....	11
IV ALGORITHMIC SYNTHESIS.....	27
V SAMPLE CLOCK GENERATION.....	34
VI DATA POINT MANAGEMENT.....	46
VII MEMORY ORGANIZATION AND CONTROL.....	51
VIII CONCLUSION.....	57
WORKS CITED.....	60
APPENDIX.....	62
A C PROGRAM FOR ALGORITHMIC PARAMETERS.....	62
B BLOCK DIAGRAMS.....	68

LIST OF TABLES

TABLE		PAGE
I	Tabular Summary of DAWG Functional Objectives and Performance Criteria	10
II	Single Stage Behavior of the Delayed Ripple Carry Vernier Counter	18
III	Dual Stage Behavior of the Delayed Ripple Carry Vernier Counter	22
IV	Key Terms of the Algorithm	29
V	The Primary Sample Clock Spectrum	30
VI	Stage Length Determination for Cyclic Operation: The Clock Division Coefficient	30

LIST OF FIGURES

FIGURE		PAGE
1.	Vernier Counter (Carry-Save Adder)	17
2.	Example Transform of F_{out} as I Becomes Greater Than 2^{-1}	23
3.	Simple Phase Locked Loop	35
4.	Prefered Phase/Frequency Detector Logic	37
5.	VCO Delay Element	39
6.	Source for $\frac{2}{3}$ Vdd Gate Voltage to Control Clamping Transistors in Delay Element	39
7.	Current Mirror which Converts V_{con} (from Loop Filter) to Ref_p and Ref_n	40
8.	Augmented PLL Block Diagram	44

CHAPTER I

INTRODUCTION

There exist today a growing number of applications in design engineering, production testing, environmental testing and system service which require specific analog waveforms and digital patterns. Such requirements are neither satisfactorily nor easily met by the use of standard function or single purpose, custom generators.

Traditional methods of waveform generation such as direct synthesis using mix-filter-divide or programmable divide-by-N, linear phase lock loops suffer from undesirable complexity or mediocre performance and are limited to the synthesis of specific, periodic waveforms, typically sine waves.

With the maturation of medium and large scale integrated circuits over the last fifteen years these techniques have been augmented by the development of direct digital synthesis. Direct digital synthesis, which is based, generally, on periodic retrieval of predetermined amplitude values, is, arguably, not synthesis, but is rather, merely reproduction. To the user of an arbitrary waveform or function generator this is not a significant distinction and in general is an issue only in specific cases. Some applications, such as electronic warfare and counter measures, which require sophisticated

frequency hopping, may be, in fact, better served by direct synthesis techniques. However, for the majority of arbitrary waveform generation applications, including medical engineering, modal analysis and electronic engineering, direct digital synthesis techniques are entirely satisfactory. In fact, if constraints of cost and area are not deemed critical, arbitrary waveform generators may be realized using relatively high speed devices and technologies, for example, bipolar, emitter coupled logic, BICMOS, and gallium arsenide, thereby achieving the required throughput speed for even the more demanding applications.

The objective of this inquiry, within a particular set of constraints, is to extend the cost/performance envelope of direct digital synthesis techniques for the generation of arbitrary waveforms. Proceeding from problem definition, a review of the pertinent literature provides the foundation for subsequent analysis, algorithmic synthesis and systemic translation. Particular emphasis is placed on examination of the functionally critical elements of the design solution.

CHAPTER II

PROBLEM DEFINITION

In the process of intellectual creation, the extent to which a synthesized solution is both effective and appropriate is dependent on the analysis on which it must necessarily be founded. Clearly the motivating desire, stated in terms of a broad functional goal, is not, generally, a soluble statement of the problem. Rather, it is a foundation for the development of such a problem statement.

In addition to the motivating objectives, two issues influence the development of specific solutions. These are the constraints externally imposed, such as limitations on resources of various kinds, and those which are imposed by the analyst. These latter constraints are called assumptions and may be either tacit or explicit. Such a context of constraints and assumptions guides the translation of motivating desires to a statement of the problem from which a solution may be derived.

The problem statement may consist of two parts, the functional objectives and the performance criteria. Functional objectives are those aspects of a design which are descriptive of its desired behavior. On the other hand, performance criteria are the metrics by which the success of each design objective is judged.

Justification is the process by which a set of design objectives and performance criteria are reviewed to ensure that they accurately reflect and address the desires which motivate the problems solution. Following justification the problem is stated in summary form.

As the need for high precision function generation and custom waveforms has grown the range of applications for which the arbitrary waveform generator is the best stimulus tool has come to include many disciplines in science and engineering. These areas include, for example, biomedical engineering, where arbitrary waveform generators may be used to simulate the heart beat or other nerve or muscle stimuli, or in zoological research where they may simulate animal calls, as well as other sounds. Modal testing encompasses a broad field of applications. Materials testing uses a variety of driving signals which may be digitally derived. An analogue of this application is vibrational simulation and testing in mechanical engineering.

A rich area of application is in the field of electronic design and testing. The output of sensors, detectors and amplifiers may be either simulated, or stimulated for test, by custom waveforms. The use of arbitrary waveform generators to stimulate electro-mechanical device cycles, for example, disk or tape drive signals, may obviate the need for application specific, custom generators to produce proprietary waveforms. Captured signals may be modified and reused in the stimulation for test of various specific circuits such as deglitching circuits or to determine, for example, circuits noise

margins. A frequency swept waveform would be suitable for filter testing.

Within the limits imposed by the system's maximum sample rate and the Nyquist criteria, any waveform may be produced using direct digital synthesis techniques. Thus, not only are all standard functions available, but waveforms for all of the foregoing applications as well.

In a sense, research and development are cyclical and interdependent. On one hand, research develops new methods, materials and devices, while development applies research results to new systems and brings to research requests for new methods, materials and devices. The goal of this research is to significantly enhance the arbitrary function generator by reducing cost and extending performance. The factors which will be minimized in order to reduce overall cost are board area, package count, use of expensive technologies, for example, ECL, and the number of application specific integrated circuits used. High speed memory should not be required. While reducing system cost, it is also expected that arbitrary waveform generator performance will be enhanced, particularly in the areas of output bandwidth and signal purity.

There are a number of constraints which will influence development of the problem statement for the digital arbitrary waveform generator or DAWG.

Research and development time is critical because of the cost of engineering time and resources, and because of the need to meet

time-to-market requirements for the product which is to be the host platform for the DAWG option. Approximately six engineer-months has been allocated for research, development, simulation, fabrication and testing.

Application specific integrations will be forged in 1.5u CMOS technology using Tektronix's Advance Development Group's standard cells for non-custom logic blocks. A single ASICs will implement all DAWG functionality except waveform datapoint memory, digital to analog conversion, and post-conversion filtering using an industry standard process.

Other than integrating, as a single ASIC, all functional blocks, with the noted exceptions, there are two ways to reduce system cost. These are to reduce the total number of parts and to avoid costly technologies and devices. In order to achieve a reduced part count preference will be given to techniques that do not require redundant architectures or high speed memories. Further reductions in cost may be gained by rejecting schemes which rely entirely on the throughput advantages offered by faster, but more expensive, logic families such as emitter coupled logic, or ECL.

A number of assumptions are made regarding the problem to be solved.

Direct digital synthesis techniques will be applied to create a system which will meet the needs of most of the application classes exemplified above. Access to a useful set of waveforms, including those of complex symmetry, indicates the use of memory dedicated to their storage. The entire AWG will be integrated as a single chip

requiring a minimum of glue logic. The advantages of system integration on a chip are well known and include savings of board area, part and manufacturing cost, and enhanced system reliability. This assumption follows directly from the resource constraints listed above.

It is assumed that this research will not treat the question of output conversion and filtering specifically. To the extent that output sample rates effect DAC selection and filter design they are considered; however, neither DAC design and their characteristics nor filter design are within the scope of this enquiry.

Techniques developed here, although generally applicable to the process of arbitrary waveform generation, will be instantiated in the context of a Motorola 68020 microprocessor controlled system. Although the DAWG operates within the context of a microprocessor controlled system, once the generator is programmed, primed and enabled, it should operate without further mediation by the microprocessor.

The primary function of the device described above must be to periodically deliver to the digital-to-analog converter an n-bit word, representing some signal amplitude. Beyond this, clearly, we must either calculate each value on the fly or recall it from some storage location. Since arbitrary waveform generation will certainly require the retrieval of predetermined datapoint values, computation on the fly is of little value in solving this problem. The specialized hardware, such as arithmetic logic units, required for such computations, while reducing memory requirements, are used for

techniques generally applicable to trigonometric functions only, and therefore are not particularly useful in the case of arbitrary waveform generation.

A digital arbitrary waveform generator will provide all standard functions, and periodic and aperiodic custom waveforms, including pseudorandom noise. Desirable operating modes include free running, gated, triggered, burst, swept, and modulated. In addition, facilities should be provided to control amplitude scaling, polarity and offset of the output waveform.

One factor controlling the versatility, and hence the usefulness, of the DAWG is the capacity and flexibility of the waveform datapoint memory. Clearly, not only must memory depth be adequate, for a given sample rate, to provide a waveform of useful length, but memory access control must also exist which allows a number of waveform representations to be stored and accessed individually or combinationally. Waveform segments, represented as blocks in memory, should be accessible randomly and repeatedly in order to easily construct a complete waveform packet. Such packets may then be combined to form a single period of the desired waveform. Waveform packet control should be mediated by the DAWG, requiring that memory organization, which would otherwise be a system level concern, be considered in this context.

Comparing desired function with the stated constraints and assumptions yields the following list of design objectives: standard functions such as sine, cosine, rectangular, triangular, and sawtooth waves; arbitrary, user defined or captured, waves; extensive memory

capacity; random access to waveform segments; datapoint segment sequencing and looping; free running, gated, triggered or swept modes; greater than 60 dB signal to noise ratio, or SNR; and an output frequency, for simple, highly symmetric waveforms such as sine, in excess of thirty megahertz. Performance objectives for SNR and maximum output frequency have been chosen to meet or exceed those of currently available devices. Various forms of modulation, as they would require an additional channel, have been rejected as being a violation of constraints on part cost and design complexity.

The design will employ techniques of direct digital synthesis, be rendered in 1.5 micron CMOS technology on a single integrated circuit chip, and be controlled by a Motorola 68020 microprocessor. Digital to analog conversion and post-conversion filtering and attenuation are not treated. In order to ease the problem of post conversion filtering the minimum sample rate will be twice the Nyquist frequency.

The object of this inquiry has been stated to be the extension of the technique of direct digital synthesis in terms of enhanced performance and reduced cost. In practical terms this amounts to application of the methods of large scale integration to the arbitrary waveform generation system in order to minimize part count and printed circuit board area while insisting that implementation be in relatively low cost CMOS technology. Performance enhancement, on the other hand, is cast in terms of bandwidth, resolution, and accuracy. In order to achieve these goals compromises have been made with respect to some functional modes. No attempt will be

made to provide for modulation capability on chip. The principal reason for this decision is the additional burden of redundant hardware and design complexity. The concomitant cost is loss of applicability to some areas of the communications field. Of course, development of a technique does not necessitate realization of all possible functions initially, nor is on chip synthesis of a modulated signal the only way such a signal may be produced.

The following table summarizes the functional and performance objectives which are the goals of this effort.

TABLE I
SUMMARY OF DAWG FUNCTIONAL OBJECTIVES
AND PERFORMANCE CRITERIA

FUNCTIONS:	MODES:	PERFORMANCE CRITERIA:
Standard Functions	Free Running	Resolution $\geq 0.005\%$
Playback	Gated	Accuracy $\geq 0.01\%$
Periodic	Triggered	Sample Rate $\geq 100\text{MHz}$
Aperiodic	Burst	Bandwidth $\geq 30\text{ MHz}$
Noise	Combination	Memory $\geq 256\text{K}$ words
		SNR $\geq 60\text{dB}$

Other considerations in the implementation of any advanced system include simplicity of design, silicon area reduction, implementation of recently matured device methodologies and the avoidance of patent or copyright infringement.

CHAPTER III

SURVEY OF SELECTED LITERATURE

Waveforms, whether simple, standard functions or signals highly complex, have been, and will continue to be, important in many fields of electrical engineering. Some of the applications which justify this statement have been mentioned in previous chapters. Digital technology, and particularly the achievement of high density integration has had a profound influence on signal synthesis.

As late as 1973 direct frequency synthesis was considered the most appropriate method for the synthesis of arbitrary frequencies, [Kroupa, 1973]. At that time the field had matured and was expressed as general theory which was soundly based in mathematics.

As digital integrated circuits became widely and inexpensively available in the mid to late seventies new techniques were devised to generate desired waveforms. capitalizing on the integrated circuits' advantages: compactness, low cost, and ease of system design, these techniques appeared in published surveys only two years latter than Kroupa [Tierney, 1975].

The technique described by Tierney, and commonly known as the phase-accumulation/lookup-table method, with various

modifications and embellishments, is still the accepted method for direct digital synthesis.

To a significant degree the appeal of the phase accumulator/lookup table technique is due to its inherent simplicity. The unit circle is divided by a number of points, each point defining a relative phase angle. The amplitude of a function at any such phase angle may be computed and stored in memory. The output frequency of the cyclic waveform is proportional to the rate at which the memory pointer cycles around the unit circle. Starting with an arbitrary phase angle in an accumulator, a fixed increment is added to the accumulator as each point is read. For a unit increment the output period of the generated waveform is the product of the number of points on the unit circle and sample rate. Higher frequencies are obtained by using increments greater than one. Lower frequencies may be obtained by utilizing increments which are less than one.

Sampling at twice the Nyquist frequency for sine, which has two points of inflection, requires four points per cycle. It follows that the maximum output frequency for waveforms of like complexity is one fourth the sampling rate. Let F_s' represent a sample rate, then:

$$f_{\max} = \frac{F_s'}{4}$$

and:

$$T_s' = F_s'^{-1} = \frac{t_{\min}}{4}$$

The most straight forward implementation of this scheme employs an accumulator with A bits, inherent power of two resolution. Since such a register will overflow naturally, additional circuitry is not necessary. By adjusting f_{\max} upward some T_s may be found which allows A to take an integer value

The lowest frequency which may be produced is:

$$f_{\min} = 2^{-A} F_s$$

and:

$$A = \lceil \log_2 (F'_s t_{\max}) \rceil$$

Note that F'_s represents an approximate sampling period based on a desired f_{\max} and that F_s represents the sampling frequency required to achieve a particular f_{\min} and frequency resolution given an A bit accumulator.

Having determined accumulator length it is now possible to calculate the sampling frequency, F_s , by the relationship:

$$F_s = 2^A f_{\min}$$

Let 2^N represent the number of points which represent a waveform. The number of bits required to address the lookup table, of length m , is N , the N most significant bits of the accumulator.

The number of bits used to represent an amplitude at a particular phase angle is a function of the desired signal to noise ratio, or SNR. SNR is defined as a ratio of power of the desired frequency to power in any other 100Hz band. This ratio is a function of the maximum amplitude error, which, for a p bit representation, is 2^{-p} .

Thus:

$$\text{SNR} = 20 \log_{10} 2^{-p}$$

and:

$$p = \lceil -\log_2 10^{\text{SNR}/20} \rceil$$

Key to the practicality of this method is the notion that all 2^A points need not be stored. If the maximum difference in amplitude between any two points is less than 2^{-p} , then either point will adequately approximate any intermediate point. The implication of this observation is critical. It follows that f_{\min} as well as the frequency resolution of the system are functions of accumulator length, A , but that memory size need never exceed 2^N . The truth of this assertion is made clear by the observation that if it is required that:

$$\sin \frac{2\pi}{2^N} \approx \frac{2\pi}{2^N} \leq 2^{-p}$$

then:

$$N = p + 3$$

The foregoing observation allows the use of memory of a practical size. Further memory reduction is based on quadrantal symmetry, that is, only a quarter sine table need actually be stored. The size of memory may be reduced further, but at the cost of the additional overhead of computation. Such schemes require multiple tables. In the simple case one table holds the sine value at a number of course points while the second holds adjustments, represented as two's complement values, which, by a complex multiply, produce the desired amplitude value [Tierney, 1975]. These, and similar tricks

are used to reduce required memory size. Given manageable memory size requirements, and the fact that several standard functions, such as sawtooth and triangular waves may be derived from the addressing sequence of the sine table, this method of direct digital synthesis is attractive in a number of ways. The foremost advantages include fast phase and frequency switching, arbitrarily fine frequency resolution, arbitrary initial phase angle, and that the entire circuit may be integrated monolithically.

A current design example which both utilizes the phase accumulator/lookup table technique and derives its sample clock from a fully digital phase locked loop has, by relaxing sample rate requirements, reached output frequencies in excess of twenty megahertz [Giebel, et al., 1989].

As a solution to the problem of arbitrary waveform generation however, there are several critical drawbacks to the phase accumulator/lookup table method of direct digital synthesis. It is clear that for arbitrary waveform generation some writeable memory must be provided. Assuming a sample rate of twice the Nyquist frequency, and memory access times of from twenty to twentyfive nanoseconds implies a maximum output frequency for simple functions, such as sine, of ten to twelve and one half megahertz. This value is far below the goal of thirty megahertz discussed in Chapter II.

Obvious solutions to consider include the possibility of accessing sets of datapoint words or pipelined memory access. Unfortunately, since the points which represent the waveform may

vary from cycle to cycle, multiple data points may not be grouped at a single address. Pipelineing, while possible, would require multiple accumulators as well as duplicate memory images of the desired waveform. This overhead is not acceptable.

Further complications arise in applying the phase accumulator lookup table technique to arbitrary waveform generation. This technique is dependent on a fixed record length. In the case of adding known distortion to a signal for example, it could be desired to modify a captured waveform. Such a waveform may be of any record length. While this problem can be overcome, the additional circuit complexity would violate the operative design constraints.

A simple, alternative method comes to mind, and, in fact, is the only other method of direct digital synthesis mentioned in the literature. Using a suitable memory configuration, datapoint word groups may be recalled and individual datapoint words may be multiplexed or shifted to the digital to analog converter at some appropriate sample rate. In this scheme there are two parameters by which the frequency of the output waveform may be changed. The first is output sample rate. The second is the number of points per cycle. In the first case the parameters for the output filter will change, requiring a different filter for large changes in F_s . If the sample rate is constant and the number of points varied then datapoint values must again be computed and stored.

Early work with this method [Shwager, et al.,1982] utilized an eight bit by 1024 word shift register and a vernier counter/divider to produce a discrete sample-rate spectrum. The fixed shift register

length and a finite number of filter configurations limit the number of frequencies which may be produced from a given waveform amplitude set. Also, each such set must be loaded, separately, into the waveform shift register, further delaying some frequency changes.

Of particular interest in the scheme of Shwager, et al, which might be referred to as a datapoint staging technique, is the behavior of the vernier divider used for sample clock generation. The essential feature of this vernier divider, which is otherwise a simple ripple counter, is the segmentation of the counter length into units, each with a number of bits such that the full ripple-carry delay to the high order bit of a segment is significantly less than the period of the system's fundamental frequency.

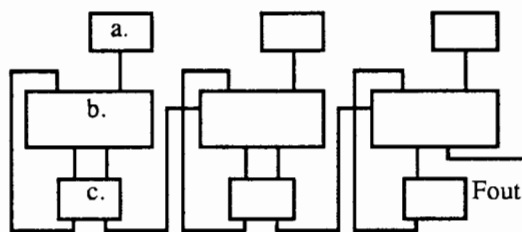


Figure 1. Vernier Counter (Carry-Save Adder).

In this counter each segment has its own increment register (a), adder (b), and sum register (c). As depicted in Figure 1, the carry out bit of the most significant segment is the source of the sample clock used in the shift register scheme of Shwager, et al.

Sum registers are latched periodically by the action of the system clock which is not shown in the figure. The effect of the

delayed ripple carry bits on the average frequency output and any resulting sample clock instability will influence the practicality of this implementation. In order to better understand this device let a single segment be examined.

First, assume the following:

The accumulator length is m bits.

$I \equiv$ increment value

Then let:

$$C_0 = 0$$

$$n_0 = 0$$

$$C_{i+1} = In_{i+1} - 2m + C_i$$

$$n_{i+1} = \left\lceil \frac{2m - C_i}{I} \right\rceil$$

$$N = \sum_{i=1}^k n_i; \text{ least integer } k \ni (k > 0 \ \& \ C_k = 0)$$

Given these definitions, and the assumption that the initial carry in is zero, the average value, F_{out} , may be expressed as a proportion of the system's fundamental frequency, f_0 ;

$$F_{out} = \frac{k}{N} f_0$$

The case in which carry in is asserted is as the last but with the increment, I' , equal to $I + 1$.

TABLE II

SINGLE STAGE BEHAVIOR OF THE DELAYED
RIPPLE CARRY VERNIER COUNTER

Carry Out	Increment = 3	f_0 Cycle Number
1	0	0
0	3	1
0	6	2
0	9	3
0	12	4
0	15	5
1	2	6
0	5	7
0	8	8
0	11	9
0	14	10
1	1	11
0	4	12
0	7	13
0	10	14
0	13	15
1	0	16

A simple example is illustrated in Table II. Let $m = 4 \Rightarrow 2^m = 16$, and $I = 3$. Then:

$$n_1 = \left\lceil \frac{16-0}{3} \right\rceil = 6 ; \quad C_1 = 18 - 16 + 0 = 2$$

$$n_2 = \left\lceil \frac{16-2}{3} \right\rceil = 5 ; \quad C_2 = 15 - 16 + 2 = 1$$

$$n_3 = \left\lceil \frac{16-1}{3} \right\rceil = 5 ; \quad C_3 = 15 - 16 + 1 = 0 \quad \text{So Stop!}$$

$$k = 3 ; \quad N = \sum_{i=1}^3 n_i = 16 ; \quad F_{\text{out}} = \frac{k}{N} f_0 = \frac{3}{16} F_s$$

Recall that F_{out} is an average value. This example shows that $3 \cdot T_{\text{out}} = 16 \cdot T_s$; however, note that the periods of the k cycles of F_{out} , listed in column one above, are not of equal length. The value of each n_i represents the number of periods of F_s in the i th cycle of F_{out} . The values of all n_i will be equal only for values of I equal to 2^{-q} ; $q \in \{1, 2, 3, \dots\}$. For a vernier counter comprised of ' s ' segments, each of length ' m ', the limit of the phase angle error is:

$$2\pi \frac{I-1}{I^2} Q \text{ rad.}$$

$$Q \equiv \frac{F_{\text{out}}}{f_0}$$

A second example, using two segments of four bits each and an increment of 0.10001_2 , equal to $\frac{17}{32}$, illustrates the significance of increments greater than one half in determining F_{out} . Let s be defined as the number of vernier register segments and assume that all ' s ' register segments are of the same ' m ' number of bits in length. Then the product, $L = sm$, is equal to the vernier counter's register length. If it is assumed that the output sample rate, F_{out} , is proportional to the fundamental clock rate, f_0 , then $F_{\text{out}} = Qf_0$. The $L-1$ bits of the accumulator register, each denoted C_i , exclusive of the most significant bit, or MSB, represent a fractional value that determines the ratio of F_{out} to the fundamental, f_0 . This ratio, Q , may be expressed in two parts. The fractional value of the $L-1$ least

significant bits, which comprise the first term, including sign, may be represented as:

$$G = -1^{I_{MSB}} \sum_{i=2}^L C_i 2^{-i} ;$$

$$C_i \equiv \begin{cases} 0 & \text{if bit 'i' is zero} \\ 1 & \text{if bit 'i' is one} \end{cases}$$

and the value of the second term may be expressed as:

$$H = 1 - 2^{-I_{MSB}}$$

Thus the ratio of F_{out} to f_0 , or Q , may be expressed as the sum of G and H :

$$Q = -1^{I_{MSB}} \sum_{i=2}^L C_i 2^{-i} + 1 - 2^{-I_{MSB}}$$

In this case, in which I_{MSB} is one,

$$Q = -\frac{1}{2^5} + \frac{1}{2} = \frac{15}{32}$$

This results from the fact that such a frequency divider may never output a frequency greater than $\frac{f_0}{2}$. In the cases in which I_{MSB} is zero, Q is the sum of all components. In the cases in which I_{MSB} is one, Q is the positive difference between 2^{-1} and the sum of all components of lesser value.

$$Q = 2^{-1} - 2^{-5} = \frac{15}{32} \Rightarrow F_{out} = Qf_0 = \frac{15}{32} f_0$$

TABLE III
 DUAL STAGE BEHAVIOR OF THE DELAYED
 RIPPLE CARRY VERNIER COUNTER

F _{out} Cycle Number	Carry Out B	I _B = 8	Carry Out A	I _A = 8	f ₀ Cycle Number
	1	0	1	0	0
	0	9	0	8	1
1	1	1	1	0	2
	0	10	0	8	3
2	1	2	1	0	4
	0	11	0	8	5
3	1	3	1	0	6
	0	12	0	8	7
4	1	4	1	0	8
	0	13	0	8	9
5	1	5	1	0	10
	0	14	0	8	11
6	1	6	1	0	12
	0	15	0	8	13
	0	7	1	0	14
7	1	0	0	8	15
	0	8	1	0	16
8	1	1	0	8	17
	0	9	1	0	18
9	1	2	0	8	19
	0	10	1	0	20
10	1	3	0	8	21
	0	11	1	0	22
11	1	4	0	8	23
	0	12	1	0	24
12	1	5	0	8	25
	0	13	1	0	26
13	1	6	0	8	27
	0	14	1	0	28
14	1	7	0	8	29
	0	15	1	0	30
	1	8	0	8	31
15	1	0	1	0	32

An interesting property of this counter is that, in the case in which the increment value is greater than 2^{sm-1} , the order and the polarity of the cycles of F_{out} will occur as if the waveform had been rotated 180 degrees about its center point as shown in Figure 2.

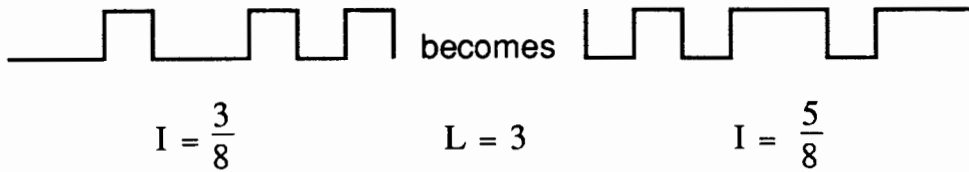


Figure 2. Example Transform of F_{out} as I Becomes Greater Than 2^{-1} .

This indicates that a judicious use of an increment value may be useful in controlling duty cycle of the divided frequency, F_{out} . This may be useful if, for example, latches are to be transparent for a fixed period.

Of course, any other sample frequency spectrum, however produced, may be substituted for that of the vernier counter. Other implementations of the staged data point technique have avoided the divider ripple delay by limiting the divisor to a non-positive integer power of two, resulting in a worst case carry distance of a single bit. This choice imposes unnecessary restrictions on the flexibility of the waveform generation system. A particular problem is that, compared to the vernier divider, which can produce frequencies of from $\frac{1}{2} f_0$ down to a minimum of $\frac{1}{2^{-L}} f_0$ in increments of $\frac{1}{2^{-L}} f_0$, the simple ripple counter/divider, restricted as mentioned above, has a

resulting sample spectrum resolution which is problematic for datapoint computation and generated frequency spectrum resolution.

Improvements to the simple datapoint staging of Shwager, et al, which uses a fixed number of data points to represent an integer number of cycles, have been in the area of memory organization and control. Real time datapoint retrieval from memory and the higher sample rates employed in this type of system require retrieval of several datapoints at once. Let these several datapoints be referred to as a group, and let the term play apply, in this context, to the process of presenting datapoints, by some means, to a digital to analog converter and post conversion filter system. Then a set of these groups of datapoint words is played one or more times, and a set of these sets is cycled over one or more times. Such control over datapoint wave information allows access to several sets of datapoints, useful in order to cover with greater resolution, a range of frequencies, or to construct a composite waveform from component packets. This has significantly improved the usefulness of the datapoint staging technique for arbitrary waveform generation.

There are several problems associated with the datapoint staging technique. Changes in effective sampling rate require that multiple filters be available. Also, unlike the phase accumulator/lookup table, or PALT method, which uses only one set of datapoints to represent a single cycle of the desired waveform at any frequency, the staged datapoint technique may need several such sets to produce a desired set of output frequencies.

On the other hand, the advantages of the technique include the attainment of the theoretical maximum output frequency of one quarter F_s , limited only by the maximum clocking rate for the technology in use. This is in contrast to the PALT method for which the output frequency is practically limited, in the simple implementation, by the required memory access time, to one quarter of the maximum memory access frequency. Other advantages include a relatively low part count, suitability for silicon integration, and flexibility of waveform representation.

It is clear from the forgoing review of these techniques that the PALT method, while offering advantages for standard functions, is severely limited in the context of the generation of arbitrary waveforms or captured waveform playback. Recall that this is due to fixed record length, point skipping irregularities, and minimum memory access time. The DpS method, by contrast, has no inherent disadvantages for the generation of arbitrary waveforms, although high sample rates require datapoint group retrieval in order to pipeline memory access. In the case in which a captured or calculated waveform datapoint set does not coincide with a group boundary, multiple cycles may be used. This requirement is eased by the use of a large, dense, discrete spectrum of sample frequencies.

Such spectrums may be produced in a number of ways other than the simple binary divider or the vernier divider of Shwager, et al.

CHAPTER IV

ALGORITHMIC SYNTHESIS

The essence of the datapoint staging, or DpS method, is that the period of the output waveform is a product of the number of points, m , used to represent a cycle and the period of the sample clock, t_s . If it is assumed that a fundamental clock frequency, F_c , may be divided by a factor, k , then the period of the output waveform, T_w , is:

$$T_w = mkT_c$$

This expression is the foundation of a solution for the problem of arbitrary waveform generation as defined in Chapter II. Given a discrete spectrum, Sc , which contains an adequate number of sample frequencies, such that $T_c^{-1} \in Sc$, then values for k and m may be found for which mkT_c is arbitrarily close to T_w . What then is the relationship between the bandwidth and granularity of Sc , and the bandwidth and granularity of the producible output spectrum? Recall the filtering advantage afforded by sampling at a minimum of twice the Nyquist rate. Adherence to this standard limits the minimum value of mk to four. This implies a worst case granularity for the output spectrum of one fourth the granularity of

S_c , and a maximum output frequency of one fourth F_c . This applies, of course, only to simple waveforms which has two points of inflection. In general, the minimum value of m , m_{\min} , for a waveform with I points of inflection, is twice I .

$$m_{\min} = 2I$$

Amplitude error is limited to finite register length induced representational errors for all frequencies for which $m, k \in \mathbb{N}$. The minimum value of m required for waveform reconstruction is $m > I$, I equal to the number of points of inflection per cycle. This is the Nyquist criteria. In general then, the maximum waveform frequency may be expressed as:

$$F_{w\max} = \frac{F_{c\max}}{2I}$$

This method, at the cost of memory access latency and some control complexity, offers improved waveform fidelity and versatility, since any wave with complexity $i \leq m-1$ may be represented in shift memory and may be produced at any $F_w = \frac{F_c}{m k}$, as specified above.

The logical extension of this approach is to attempt to control both m and k dynamically. In the case of m this amounts to selectable shift register length. In the case of F_c , selection from a continuous range of values is not practical. Instead, selection of the most suitable value from a discrete spectrum may be sufficient to allow outstanding performance by many currently accepted metrics.

Consider a specific example. Assume a comfortable sampling rate for 1.5 μ CMOS technology to be 100 $\frac{MS}{s}$ to 125 $\frac{MS}{s}$ and the minimum number of points per cycle (MPPC) required to reconstruct a simple waveform to be four, as is the case for simple, symmetric waveforms such as sine. Further assume that the discrete frequencies of the clock spectrum span the above range with granularity of 10E3Hz. Maximum shift register length, MSRL, in order to avoid extreme capacitance on the clock lines required for a discrete implementation, is assumed to be relatively short; m_{max} no greater than sixtyfour words, each of twelve bits. In cases calling for a greater number of data points, the system functions as a pipeline driver for the requisite data stream.

TABLE IV

KEY TERMS OF THE ALGORITHM

Datapoint Word Length $\equiv p = 12$ bits

$$F_t^{-1} \equiv F_w^{-1} = T_w = mkT_{ci}$$

$F_{ci} \equiv$ Sample Clock Spectrum Element

$F_t \equiv$ Target Frequency

$F_w \equiv$ Output Waveform Frequency

Minimum sample period $\equiv T_{c0} = 8ns$

MPPC $\equiv m_{min} = 3$

MSRL $\equiv m_{max} = 64$

$C \equiv$ PLL frequency resolution =10E3

TABLE V
THE PRIMARY SAMPLE
CLOCK SPECTRUM

$$F_{ci} = T_{ci}^{-1}$$

$$T_{cmax} \leq T_{ci} \leq T_{c0}$$

$$T_{c0} = 8ns$$

$$T_{cmax} = \frac{T_{c0} \text{ MPPC}}{\text{MPPC} + 1}$$

$$T_{ci} = T_{ci-1} + 0.64 \text{ ps}$$

TABLE VI
STAGE LENGTH DETERMINATION FOR CYCLIC OPERATION:
THE CLOCK DIVISION COEFFICIENT

$$mk = \text{int} \left[\frac{T_t}{T_{c0}} \right]$$

$$k = \text{int} \left[\frac{mk}{\text{MSRL}} \right]$$

$mk < \text{MPPC} \Rightarrow$ frequency out of range

$\text{MPPC} \leq mk \leq \text{MSRL} \Rightarrow k = 1$

$mk \geq \text{MSRL} \Rightarrow m = \text{MSRL}$

The preceding are explicit in the previous statements or may be determined from them.

An alternative approach for the last case in table VI would be to stream data through the integrated circuit in a flow mode. Thus, there are two models for the movement of data from data point memory to the analog portion of the arbitrary waveform generator.

For cases in which $m \leq \frac{\text{MSRL}}{2}$, μ multiples of the data point sets are loaded into the length μm main shift register. The total number of data points to load, and therefore the utilized shift register length, m' , is defined:

$$m' \equiv m * \text{int} \left[\frac{\text{MSRL}}{m} \right]$$

hence the total number of cycles in the main shift register is:

$$\mu = \frac{m'}{m}$$

What can be observed about the signal error? Clearly amplitude error, E_A , results from data point value truncation. For twelve bit, twos-compliment data points that error, expressed in decibels is:

$$E_A = 20 * \log_{10}(2^{12}) \cong 72.25 \text{ dB}$$

The discrete nature of the sample clock spectrum will be manifest as induced output frequency granularity corresponding to

a maximum of one half the frequency resolution:

$$R_f = \frac{C}{mk}$$

Let phase error, E_θ , be represented as the difference between the ratio of the selected sample clock frequency and F_{out} , and unity.

Then:

$$E_\theta(F_t) = \frac{F_{ci}}{mkF_t} - 1$$

For notational convenience let $Y = mkF_t$; then a simple rounding function to map Y to F_{ci} may be expressed as:

$$C * \text{int} \left[\frac{Y}{C} + 0.5 \right]$$

This then yields:

$$E_\theta(F_t) = \frac{C * \text{int} \left[\frac{Y}{C} + 0.5 \right]}{Y} - 1$$

Since the phase error will be largest for minimum value of mk , and $MPPC = mk = 4$, the values of F_t which result in maximum $E_{\theta_{max}}$ must be:

$$5^{-1}F_{c0} + \epsilon \leq F_t \leq 4^{-1}F_{c0} ; \quad 0 \leq \epsilon \leq 1$$

$$\text{ABS}(E_\theta) \text{ will be a maximum when } C * \text{int} \left[\frac{Y}{C} + 0.5 \right] = Y + \frac{C}{2}$$

Then the maximum error is:

$$E_{\theta\min} = \frac{C}{2Y} \cong 5E-5 \text{ as } \epsilon \rightarrow 0 \text{ when } F_t = 5^{-1}F_{c0} + \epsilon$$

Practically, this means that a target frequency, f_t , in the range 25E6 Hz to 25,001,249 Hz will be generated as 25E6 Hz, those in the range 25.00125E6 Hz 25.0025E6 Hz as 25.0025E6 Hz. This is the result expected: a worst case granularity of $\frac{10\text{kHz}}{4}$, or 2500Hz. This analysis does not take into consideration reference spectrum frequency irregularities. Such irregularities may, for example, result from PLL instability and the influence of systemic noise.

The DpS method accepts some additional post conversion filter overhead and output frequency granularity for an exact technique for sampled wave form generation. The enhanced high frequency capability, for a given technology, recommends this approach. PALT methods can not approach this fidelity at the upper end of their frequency range in a conventional CMOS implementations.

Analysis of this algorithm reveals three critical elements which must be treated if the DpS technique is to be implemented successfully. First is that of sample clock spectrum generation. Even if it is assumed that a primary sample clock spectrum, of granularity C is available, the use of a simple binary counter/frequency divider to achieve $\frac{F_{ci}}{k} = kT_{ci}$ implies a gap between frequencies, at the top end of the sample clock range, equal to $\frac{2F_{c\min} - F_{c\max}}{2}$. Such a broad range of frequencies, if left uncovered, would substantially reduce the usefulness of this method. The second area of concern deals with the issue of sample staging in cases in which m exceeds

MSLR. Related to this question, but also of general significance, is that of how waveform memory is to be organized and controlled. The solution of these last two problems must be compatible. Each of these issues will be addressed in following chapters.

CHAPTER V

SAMPLE CLOCK GENERATION

Implementation of the algorithm developed in Chapter IV requires the generation of a discrete sample spectrum, made up of elements, F_{ei} , which are effective sampling frequencies related to F_{ci} as:

$$F_{ei} = k^{-1}F_{ci}$$

Any time a clock generator is expected to function near the limits of a particular technology, to provide a finely spaced, discrete spectrum of sample frequencies may require the use of a phase locked loop, or PLL. The carry-save adder solution employed by Schwager, et al. is a very good one but is not the only acceptable approach. In particular, the use of the carry-save adder method limits the maximum sample frequency to $\frac{1}{2} F_c$.

Advantages to the use of a PLL include: good accuracy and self regulation; the output frequency may be selected by use of programmable prescalars; the application of prescalars to either the input or feedback path allow the PLL to be used as either a frequency divider or multiplier, respectively.

The first solution considered was the use of a shift/increment divider, to be described later, in conjunction with a tunable phase

locked loop. Such a module would generate F_e .

The purpose of the PLL is to match the phase and frequency of its possibly divided output signal to those of its stimulus. One early application of the digital PLL was in the tuning sections of citizen band transceivers [Breeze, 1978]. Early work describing the charge-pumped PLL and an analysis of the concomitant phase error was done by Gardner [Gardner 1980,1982]. A subsequent attempt to produce a tunable, digital phase locked loop, implemented in discrete components, was reported from the University of Washington [Dahmani, 1981]. Dahmani's system operated over a frequency range of from 140Mhz to 170Mhz, although a target resolution of 5 kHz was not achieved. More recently CMOS PLLs have been used as the basis for arbitrary waveform generators as well as numerous CMOS clock generation circuits [Caviglia, et al. 1983].

The principles of the PLL are well understood and have been presented numerous times. Therefore, only a sketch of the idea is presented here.

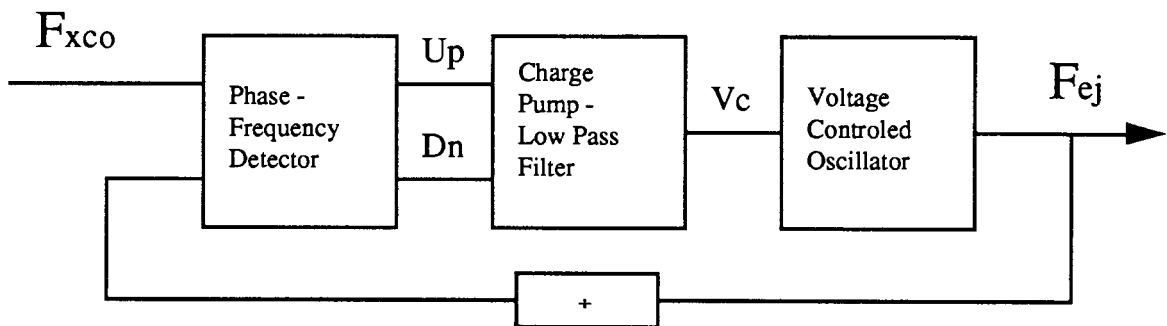


Figure 3. Simple Phase Locked Loop

The concept of PLL operation is simple . Assume, for the moment that the divider block is set to unity. The phase/frequency detector, or PFD, measures the time between a rising(falling) edge of an input signal to the corresponding rising(falling) edge of the other input signal. This results in an output pulse with duration proportional to that measured difference. This pulse will occur on one of the two outputs of the PFD, depending on whether the oscillator output is fast or slow. If fast, charge is drained from the control node; if slow, charge is pumped onto the control node. The transient effects of this charge pumping are moderated by a following low pass filter. Use of a first order active filter, with transfer function $F(j\omega) = \frac{1+j\omega\tau_2}{j\omega\tau_1}$, approximates integration of charge over time at low frequencies. Thus filtered, the control node voltage determines the operating frequency of the voltage controlled oscillator, or VCO . It should be noted that the range of the feedback loop divisor, N , is limited since

$$\frac{\zeta_{\max}}{\zeta_{\min}} = \sqrt{\frac{N_{\max}}{N_{\min}}} \leq 2$$

must obtain in order that the loop filter dampening factor, ζ , be limited to a range of from 0.5 to 1.0, [Best, 1984].

Critical components of the PLL are the PFD, the divider circuit(s), and the VCO. The generally accepted PFD logic circuit [Breeze, 1978], [Jeong, 1987], [Giebel, et al. 1989] may be either rising or falling edge sensitive, depending on whether implemented with

nor or nand form logic and is shown in Figure 4. An integrated phase detector with low device count was reported from Philips Research Laboratories [Woudsma, Noteboom, 1985].

Although sensitive to both phase and frequency, this circuit will allow some uncompensated drift within a small envelope about the desired frequency. This occurs because the phase difference, $\Delta\phi$, may be smaller than some non-zero propagation delay asymmetry associated with the latch elements of Figure 4. This propagation delay differential is referred to as the dead zone [Breeze, 1978] and results in some reduction of the phase-error-compensatory pulse width as $\Delta\phi$ approaches zero. This effect may be minimized by appropriately sizing transistors during circuit layout, in order to balance the relevant propagation delays.

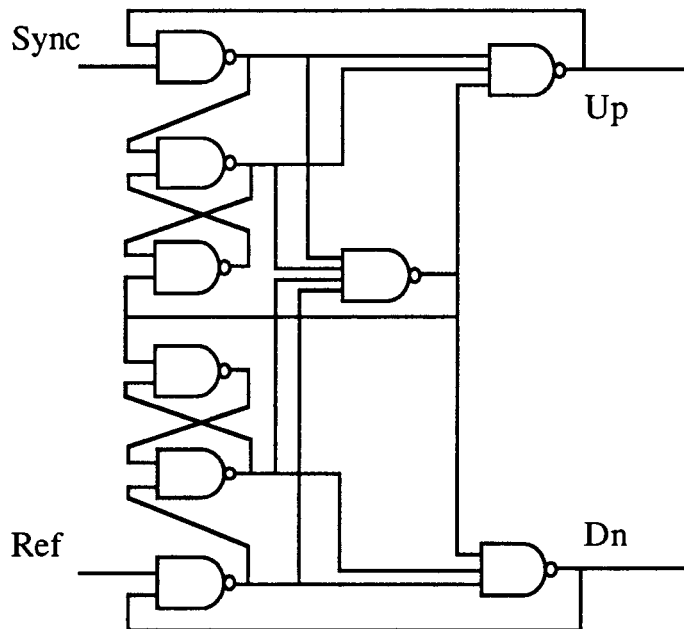


Figure 4. Preferred Phase/Frequency Detector Logic

In applications requiring improved behavior an anti-backlash circuit [Breeze, 1978] may be added, however, recent reports of clock generation schemes [Jeong, et al. 1987], based on integrated PLL implementations, have successfully employed the logic of Figure 4, without the need for dead zone effect compensation. Simulations done using both FASTSIM, a digital gate level simulator, and TEKSPICE, Tektronix enhanced version of SPICE, suggest that the simple circuit of Figure 4 will function well in this design.

In order to produce the desired spectrum from 100 MHz to 125 MHz, with granularity of 10 kHz, it is necessary, in the worst case, to divide F_{ej} to 10 kHz. Since the reciprocal of κ will always exceed ninety, any κ may be expressed as:

$$\kappa = (11p + 10q)^{-1}$$

A device with such a dividing coefficient is known as a dual moduli divider, a specific and simplified form of the more general shiftregister divider, or SRD. One possible implementation would use a 11/10 length-selectable shift register with a pair of associated down counters and glue logic to toggle a shiftregister length control bit and reload the counter when terminal count is equal to zero. Such a device was designed and simulated using FASTSIM. Results indicated outstanding performance from this circuit.

In the general case a shiftregister divider is composed of an increment register, an accumulator, and a shiftregister of selectable length. The minimum shiftregister length is constrained to be at least long enough so that the worst case ripple delay of the accumulator is

less than the minimum shift cycle period. For a minimum shiftregister length of M , the maximum F_{out} is $\frac{1}{2M} F_{in}$. The minimum F_{out} and the step between frequencies is fixed by the length of the ripple divider, n , as well as M .

$$F_{min} = \frac{1}{2^n M}$$

The final, and perhaps most critical, element of the PLL is the voltage controlled oscillator. Recent implementations of digital PLLs have employed ring oscillators using various delay elements. Some of these simple ring oscillators employ current mirror links to a filtered control node, effecting propagation delay by restricting the inverter's source and sink currents [Jeong, et al. 1987]. TEKSPICE results for this configuration were unsatisfactory in terms of the available control sensitivity verses propagation delay. A more recent report [Giebel, et al. 1989] indicates that differential inverter pairs, controlled by a mirrored current source, and with clamping transistors to minimize the voltage swing, as well as supply induced frequency fluctuations, may be used to build a very stable ring oscillator. To avoid some of the complexity of the Geibel delay element a single sided version was designed.

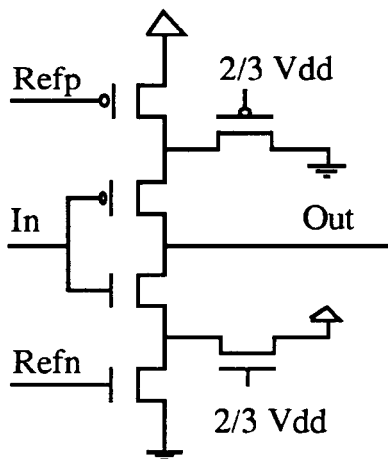


Figure 5. VCO Delay Element

This configuration differs from that of Geibel et al. in that, in addition to being single sided, it uses a gate voltage of $\frac{2}{3} V_{dd}$, rather than $\frac{1}{2} V_{dd}$, for the clamping transistors. The required $\frac{2}{3} V_{dd}$ is produced using the circuit in Figure 6.

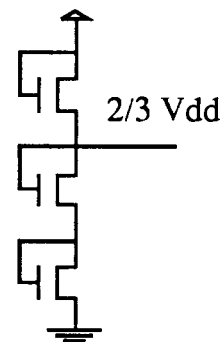


Figure 6. Source for $\frac{2}{3} V_{dd}$ Gate Voltage to Control Clamping Transistors in Delay Element.

The charge pumped and filtered potential of the control node, V_{con} , controls the propagation delay, and, hence the frequency, of the ring oscillator. Recall that the delay element has two control inputs, Ref_p and Ref_n . The potential V_{con} is in fact Ref_n . In order to generate Ref_p a simple current mirror is used as shown in Figure 7.

These structures, properly sized, provide approximately linear control of F_{ej} versus V_{con} , with a control sensitivity of about 10 kHz per 600 microvolts. These simulation values vary somewhat, depending on process and device swings.

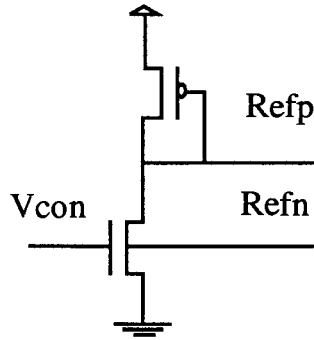


Figure 7. Current Mirror which Converts V_{con} (from Loop Filter) to Ref_p and Ref_n .

The associated process space may be visualized as having three dimensions: general process characteristic; P-type device characteristics; N-type device characteristics. Thus, the dies resulting from a particular process may exhibit behavior ranging from the slow/slow/slow corner at one extreme to fast/fast/fast at the other. Of course, beyond statistical methods, there is no way of knowing where in the process space a particular wafer might fall. In order to enhance fabrication yield it is desirable to use as much of this space as possible. Iterative techniques were used in device sizing of the

VCO components. This reduced the effects of movement into extreme parts of the process space such as slow/fast/slow. Even though some effects of process, such as control sensitivity and linearity, were moderated by careful device sizing, the gross effects of process speed on the number of elements required in a chain, to achieve a particular range of output frequencies, could not be fixed. The obvious solution to this problem is to provide a chain of delay elements adequate for the fast/fast/fast corner of the process space and to then make the chain length selectable. Provided that the control range is greater than a single element delay there would be no gaps in the output range of the ring oscillator. The configuration described above has been extensively evaluated using TEKSPICE and covers the process space with delay cycles of length five to thirteen elements.

Simulation and evaluation for the PLL as a whole has been difficult. The circuit model is cumbersome due to the large number of devices represented as well as the relatively long simulation time needed to observe pull in. The following method of simulation was employed. Assume that V_{CON} is a specific value. This value of V_{CON} will produce an output frequency from the VCO; call it F_1 . Further assume that the desired frequency, F_w is close to F_1 . It is now possible to calculate a Δt representing the appropriate lead/lag relationship of the simulated output of the PLL to that of an assumed reference frequency, F_{XCO} . This Δt may then be used as the basis for stimuli for a second simulation, that of the PFD and the Charge Pump Loop Filter, or CPLF. The output of this simulation is a V'_{CON} which

then replaces the first, assumed V_{con} . The sequence is then repeated until Δt is practically zero. These simulations indicate effective locking of the loop.

Given a tunable PLL, consider again the problem of sample spectrum breadth and granularity. The problem solved by the vernier counter, otherwise known as the carry save adder [Westi, et al. 1985] is of no concern if the worst case ripple delay is less than the fundamental frequency. The Schwager implementation samples at a maximum rate of $\frac{F}{2}$; therefore, its maximum sample rate will be one half the technologically imposed maximum. Alternatives include the use of well known and understood high speed adder techniques such as binary look-ahead or carry select adders. Since the ultimate limitation on output sample rate, for a given technology, is the ability to drive amplitude values off chip it seems reasonable to believe that appropriately segmented adders of such design would be able to accommodate clocked addition at rates of F_{ej} , at or near this technologically imposed maximum through put rate, in the case of CMOS. The complexity of these adder schemes, on the other hand, is likely to be unduly expensive in terms of design time and area. After all, the object is to divided a frequency not to implement a high precision adder.

If it were possible to pre-divide the frequencies of the primary sample clock spectrum to some frequency such that the period of that frequency were greater than the worst case ripple delay of the slower ripple counter/divider then the ripple carry

delay is not an issue. Division by four allows a ripple carry delay of thirty-two to forty nanoseconds. Such a period allows for a divider length of greater than 20 bits in the case of 1.5μ CMOS technology. This indicates that a four bit shift chain might be used for the initial division, with further division achieved by means of the simple ripple counter/divider. This is the shiftregister divider described above.

As mentioned earlier, use of small positive integer divisors results in large gaps at the upper end of the generated spectrum. That is, given a primary spectrum as discussed previously, ranging from 100 MHz to 125 MHz, integer values of κ equal to two, three, and four will produce gaps in the secondary sample clock spectrum of $37\frac{1}{2}$ MHz, $8\frac{1}{3}$ MHz, and $2\frac{1}{2}$ MHz respectively. Such broad gaps in the secondary sample clock spectrum would significantly handicap the DpS technique. These gaps may be filled by appropriate extension of the selectable length delay element ring of the VCO. This results in a broadening of the primary frequency spectrum. For example, assume that a sample clock frequency of 35 MHz is desired. From a fundamental of 105 MHz division by three is achieved using a shiftregister divider. Note that in this case the variable length shift register is the only element of the SRC required. For a counter/divider of n bits and a minimum shiftregister length of five a spectrum of from $(2^n * 5)^{-1} F_{ci}$ up to $(2 * 5)^{-1} F_{ci}$ may be produced with granularity equal to $(2^n * 5)^{-1} F_{ci}$.

The clock generation scheme described above is an alternative to the use of a simple ripple counter/divider, constrained to power-of-two divisors, and, in conjunction with a sufficiently high fundamental would be quite adequate for this application. However, since the maximum output frequency is $\frac{1}{10}$ to $\frac{1}{8}$ of F_{XCO} the output frequency for F_{ei} is limited to 10 MHz to 12.5MHz for F_{XCO} equal to 100MHz.

The PLL, shown below in Figure 8, will improve flexibility in terms of sample clock spectrum breadth and granularity.

This configuration allows for great flexibility in the range of

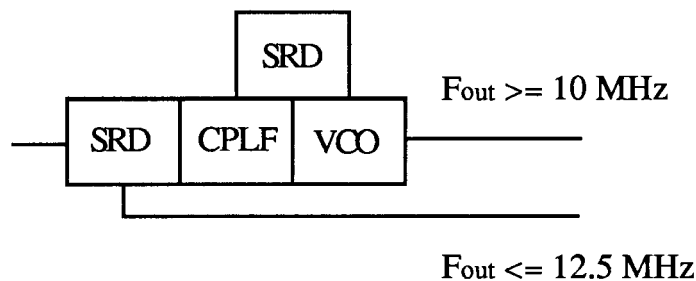


Figure 8. Augmented PLL Block Diagram

useful values of F_{XCO} , the range of VCO output frequencies, and VCO output division. By virtue of a selectable VCO delay cycle length, both process-skew-effects compensation and a tunable frequency range may be achieved. If the maximum selectable length is increased from thirteen to thirty-one the tunable range is increased from 100 M to 125 MHz to 10 M to 125 MHz. The granularity for this part of the

sample spectrum is expected to be 10 kHz. Sample rates below 12.5 MHz may be produced using shiftregister/divider techniques applied to an F_{xco} of 100 MHz or greater. The granularity for this part of the sample spectrum is as reported above, $\frac{1}{2^{n*M}} F_{xco}$.

CHAPTER VI

DATA POINT MANAGEMENT

The objective of data point management is the storage and movement of points of data, representing phase specific amplitudes of some waveform of interest, within the context of the DAWG. In order to accomplish this task it is necessary to provide a data point memory, or DpM, dedicated to the storage of these values, mediation of processor transfer of data values to DpM, and the recovery of these values by the DAWG when required to reconstruct a particular waveform. Note that the data points from the microprocessor may be either calculated or acquired values.

Since, in order to produce high frequency output, it is necessary to drive data off chip every eight nanoseconds, and yet expensive memory systems are to be avoided, a sufficient number of data points must be recovered during each read cycle to maintain data flow while a second read cycle is effected. This is true in both the cyclical shift mode and the flow mode which were introduced in Chapter IV. This requirement indicates that multiple data points must be located at a common address in DpM. On recovery of such a group of data points they must be properly staged for transfer to the DAC and post conversion circuits.

This system is assumed to have a dedicated chip select, a sixteen bit wide data bus to the microprocessor, and an address space of one megabyte. One half of this address space is dedicated to the transfer of data points to the DpM. Writing sixteen bytes to the bottom half of the address space allows eight twelve bit data points to be collected in the DAWG. When the low address nibble is 1110₂, indicating the transmission of the fifteenth and sixteenth byte, the address on A18...A4 is driven onto the address bus of the DpM and the collected ninety-six bits, representing eight points of data, are driven onto the DpMs dedicated data bus. The loading of DpM is handled by memory mapping the device function space to address space.

If the waveform to be generated is represented by only a few points, either due to limitations imposed because a high output frequency is desired, or because the waveform is highly symmetric, it is desirable to avoid repeated retrieval of these same data points. This feature is provided by means of a twelve bit wide, sixty-four word long shift register, which may be thought of as a waveform cache. Note that the problems associated with clock loading would become severe for shift structures much larger than that described. Assuming a nominal gate capacitance of approximately 0.2 pF, the capacitive loading on the clock due to this structure is:

$$C_{\text{gate}} = 12 * 64 * 0.2 \text{ pF}$$

$$C_{\text{gate}} = 153.6 \text{ pF}$$

The advantage to this approach is in its reduction of noise due to data-transfer-associated memory accesses and the possibility of frequency sweeps with only a change of the increment register value of a SRD.

Given an adequate range of filters, any sufficiently simple waveform may be represented by sixty-four points. This implies that up to 4096 different waveforms of this length may be stored in DpM. Since each of these waveforms must be sampled at a rate compatible with at least one filter from a finite set, frequencies below some value must require more than sixty-four points. The transition band within the device's frequency range is a function of the clock spectrum breadth and granularity, and of the number and characteristics of the post-conversion filter set. The program listed in appendix A may be used to find the frequency divisor, k , for any frequency within system range, as well as a specific primary sample period, T_{ci} . If $F_{ej} = \frac{1}{kT_{ci}}$ is a sampling frequency which is not covered by the filter set the described waveform cache can not be used.

If a target frequency and a sample frequency have been selected, and the target period exceeds sixty-four times T_{ei} , then the data must be streamed from the DpM, through an alternating pair of staging registers, the waveform cache shiftregister, and on to the DAC and filter system. This mode of system operation, referred to as the flow mode, is quite simple. On each pulse of the sample clock a data point is shifted from staging register A into the sixty-four word main shiftregister, or MSR. Concurrently, a read cycle to the DpM is

generated. This cycle will cause staging register B to be loaded with the next eight twelve bit words. After eight shifts, which, at 125 MHz, is 64 ns, the cycle ends, and the roles of register A and register B are reversed. This functional role swapping of the registers used to load the main shiftregister, or waveform cache, gives rise to the expression "ping-ponged" staging registers.

The implementation of these features is by means commonly accepted techniques for data transfer and control. Details of memory organization and its control are treated in Chapter VII.

CHAPTER VII

MEMORY ORGANIZATION AND CONTROL

Data point memory organization determines the practical use that may be made of a given number of data point storage locations. Recall that the necessity of data point pipelining, imposed by the need to reduce read latency, compels the use of data point groups of size eight. Since this octet of twelve bit words may be thought of as a single ninety-six bit data block one simple solution is a linear memory.

Linear memory would be simple to use. Start the cycle at α ; stop it at β . Any appropriately sized set of data point groups could be loaded into an available segment of the DpM. The only information needed in order to retrieve the waveform is the start address, α , and the number of points, equal to $\beta - \alpha + 1$. Unfortunately, in the worst case this would require loading of the DpM by the microprocessor every time a new waveform is desired. A further inconvenience is that this scheme allows only those sets of Dp groups which are contiguous in memory to be included in the same waveform.

A more useful method of DpM organization would be one which allowed assembly of waveforms from constituent packets. Conceptually, a packet is one or more repetitions of sets of octets.

In such a context a set of contiguous octets, representing, for example, a sine wave, might be juxtaposed with another set, representing a triangle wave or some other standard or arbitrary waveform, to create a new waveform. Free running generation is, then, a continuing cycle over a collection of packets. This collection is a concatenation of several packets. Each packet is comprised of one or more repetitions of a contiguous set of octets. Let A be defined as the number of times a set of octets is repeated in a packet. Further, let S be defined as a contiguous set of octets, thereby representing a double of start address and length. The packet, P , is:

$$P = AS$$

In the following expressions, Σ is used to indicate concatenation of packets to create a series of periodic sampled amplitudes which represent a particular waveform, W .

$$W = \sum_{i=1}^m P_i$$

Here, m is less than, or equal to, the length of the ASC RAM, and represents the cardinality of P . The number of unique entry points to the RAM table, the maximum length of a DpW sets, as well as the upper bound on A , will be fixed by the number of bits provide for their representation in the ASC RAM.

The provisions described for the control of DpM access and waveform construction have been defined so as to enhance the flexibility of waveform generation. The extent to which this goal is

achieved is dependent upon the values chosen for the free parameters of the memory access control module. For discussion a total of 256 megabytes of storage, organized as 2^{15} words of ninety-six bits each, and referred to as octets, is assumed. This waveform storage space is consistent with the target for this system, described in Chapter II.

Clearly, the words from the ASC RAM will define the packet constructed waveform. Assume that the ASC RAM is thirty-two words of twenty-seven bits. This configuration fixes m at thirty-two. Assuming that the entry points to the DpM are equally spaced, packet granularity, G_p , is a function of the number of bits used to represent the thirty-two DpM entry points, as well as DpM length. If a five bit representation is used the entry point granularity, in this example, is 2^{10} octets. Dividing evenly the remaining twenty-two bits between the packet length and packet repetitions functions would allow a set of octets to have cardinality 2^{11} , and to be repeated 2^{11} times. This indicates a maximum 2^{25} points per packet. Notice that the entire DpM would be read if $A = 1$ and for each S , $|S| = 2^{10}$. Then the number of DPWs covered is equal to $m|S|$, or 2^{15} . One advantage gained by the use of a cardinality of S which is greater than the G_p is that it allows looping over contiguous groups of sets, S_a, S_b, S_c, \dots , et cetera. Using the parameters given above, any two sets of 2^{13} data points may be cycled up to 2^{11} times, based on the contents of a single ASC word. Thus the range for the number of points in a collection of packets, which represent a waveform, is from eight data points to 2^{30} data points.

In the case of the maximum, of course, there would be segments within the waveform which were locally periodic. Again this follows from the fact that, for the assumed values, in the extreme , $2G_p = |S|$. For example, let the contents of S_a, S_b, S_c, \dots be denoted as A, B, C respectively, with $|S_i| = G_p$. The the extreme case, with 2^{30} points, may be represented as $AB_1, AB_2, \dots, AB_{2048}, \dots, YZ_{2048}, ZA_1, \dots, ZA_{2048}, AAB_1, \dots, FFG_{2048}$. In this context the cycles are of the form AB, et cetera.

Since both the ASC RAM and DpM are of finite extent there are optimal parameters for entry point granularity and $|S|$. For optimal flexibility of use it must be possible to start a packet at any octet, and, for $L =$ number of octets in DpM, the relationship $1 \leq |S| \leq L$ must obtain. Under these conditions S may start at any DpW and wrap around through the balance of DpM. Remaining free parameters include both the number of packets, equal to the number of words in ASC RAM, and the maximum number of times a particular S may be cycled for a single packet. Selection of values for these two remaining parameters is to be made on the basis of criteria which are beyond the scope of this work.

From the discussion above the significance of the ASC RAM is clear. Its implementation follows a somewhat conventional model. Access to the ASC RAM is controlled by a simple counter. Starting from some address, $1 \leq sa \leq m$, on the completion of the last repetition of some S the packet is complete and the counter is incremented. In order to load the ASC RAM the effective address is mux selected either from the counter or the chip's internal bus.

Once loaded, the ASC word at the current address is used to load the address pointer to DpM, the S count decrementer, and the repetitions decrementer. Every eight sample clock cycles the ASC module clock is toggled. This clock pulse advances the pointer into DpM, and decrements the count of unaccessed octets in S. When terminal count is reached by the S count decrementer, end of set is indicated and the repetitions count is decremented. Terminal count of the repetitions count indicates end of packet and increments the pointer into the ASC RAM.

In addition to simply controlling the recall of octets from DpM it must be possible to interrupt the process described above in two cases. Recall from Chapter IV that there are two modes of operation for the DAWG. These were the waveform cache mode and the flow mode.

In the case of the waveform cache mode a relatively small packet, representing one or more cycles of the desired waveform, may be caused to flow into the main shift register. The size of this packet will not exceed eight octets, or sixty-four data points. Any unnecessary data points may be ignored since the main shift register is length selectable from thirty-three to sixty-four.

The flow mode case, on the other hand, is more complex. Now it is necessary to fill one of the two eight word staging registers, as well as the sixty-four word main shift register. Once the staging register is filled the flow mode process must be allowed to continue only if output is qualified by the chip's mode control block. If a

trigger or a gated qualification is pending the flow of data from the DpM into the chip, and hence, out to the DAC, must be suspended.

Interruption of the loading of octets during pipeline loading is controlled by the prime/hold count decrementor. If control is in the flow mode the prime/hold count decrementor is loaded with nine. After configuration of the ASC module, when ASC RAM address control is switched to the ASC RAM address pointer, the prime/hold count decrementor's non-asserted HOLD output allows the ASC module, including the prime/hold count decrementor to be clocked. When the prime/hold count decrementor reaches terminal count the output HOLD is asserted. In the absence of a QUAL assertion from the mode control module the ASC module clock is disabled, thereby effecting the interrupt.

CHAPTER VIII

CONCLUSIONS

The success of an undertaking may be judged by the extent to which it meets its several objectives. In the case at hand the objective has been stated to be, within the constraints imposed by the ubiquitous demand for minimal cost, to develop, exclusive of the digital to analog conversion and filtering, an arbitrary and captured waveform playback device with superior performance characteristics. These characteristics were tabulated in Table I.

Review of the literature indicated two techniques for the digital generation of waveforms. These were the phase accumulator/lookup table method, and the data point shift technique. The PALT method, because of its need for non-linear memory access, requires either high speed memory, duplicate hardware, or both. On the other hand the DpS method primarily requires a highly stable, tunable sample clock. Due to the limitations placed on the PALT technique by the assumed constraints, the DpS approach was chosen for development.

An algorithm which determines an optimal number of samples per cycle and a sample clock rate, while minimizing the difference between the frequency produced and the original target

frequency was developed. The number of samples per cycle and the sample clock rate are both functions of the system's maximum number of points per cycle, the sample clock range and its granularity, as well as the target frequency.

The solution, based on this algorithm, loosely follows the work of Schwager, et al. and Caviglia, et al. The extensions offered here include dynamic selection of the effective main shift register length, provisions for a waveform flow mode, as well as an enhanced sample rate bandwidth which is over twice that offered by Caviglia. These improvements provide for improved accuracy, long, possibly aperiodic, waveforms, and an extended output bandwidth. Particularly, a new ring oscillator delay element and associated bias circuitry were developed. A feature worthy of note is the use of clamping transistors, with their gates tied to $\frac{2}{3} V_{dd}$, to moderate the behavior of the delay elements.

The two modes of the device have been described as the cyclical and flow modes. They allow, in the first case, a small set of data points representing one or more periods of a desired waveform to be cycled through a recirculating shift register at a precisely controlled clock rate. In the case of the flow mode, more complex waveforms and those of lower frequency are streamed through the part from a dedicated data point memory.

One alternative to linear organization of such a data point memory was presented. These techniques for assembling a wave from pieces, generally referred to as sequencing, was described as an

assembly of packets. Sets of octets of data points could be repeated some integer number of times to make up a packet. These packets would collectively represented a waveform.

The result of combining these several developments in a novel structure is the DAWG, or Digital Arbitrary Waveform Generator. This system, although conceptually simple, is capable of performance which matches or exceeds that of many currently available AWGs, regardless of cost. Within the constraint space assumed for this design, no similar device, of which this author is aware, can match DAWG's performance.

Developmental progress on this project continues and the patent application process is active. Fabrication is anticipated to take place in the spring of 1990.

WORKS CITED

- Best, Roland E. Phase Locked Loops: Theory, Design, and Applications. New York. McGraw-Hill. 1984. p. 219
- Breeze, Eric. A New Design Technique for Digital PLL Synthesizers. Transactions on Consumer Electronics. IEEE Vol. CE-24 No.1 February, 1978. p. 24-33
- Caviglia, D. D.; De Gloria, A.; Donzellini, G.; Parodi, G.; Ponta, D. Design and Construction of an Arbitrary Waveform Generator. Transactions on Instrumentation and Measurement. IEEE. Vol. IM-32, No. 3. September, 1983. p. 398-403
- Dahmani, Danmaned. 1981. Design of a Low Cost Frequency Synthesizer in the Range: 140 MHz to 170 MHz. Masters Thesis, University of Washington.
- Gardner, Floyd M. Phaselock Techniques. Wiley & Sons. New York. 1966.
- Garner, Floyd M. Charge-Pump Phase-Lock Loops. Transactions on Communications. IEEE. Vol. COM-28 No.11 November, 1980. p. 1849-1858
- Gardner, Floyd M. Phase Accuracy of Charge Pump PLLs. Transactions on Communications. IEEE. Vol. COM-30 No.10. October, 1982. p. 2362-2363
- Giebel, Burkhard; Lutz, Jürgen; O'Leary, Paul L. Digitally Controlled Oscillator. Journal of Solid State Circuits. IEEE. June, Vol. 24, no. 6 June 1989 p. 640-645
- Jeong, D.; Borriello, G.; Hodges, D.; Katz, R. Design of PLL-Based Clock Generation Circuits. Journal of Solid State Circuits. IEEE. Vol. SC-22 No. 2. April, 1987. p. 255-261. Figure 3.a.

- Krupa, Venceslav F. 1973. Frequency Synthesis. New York: Wiley & Sons.
- Tierney, Joseph. 1975. Frequency Synthesis: Techniques and Applications. Ed. Jerzy Gorski-Popiel New York: IEEE Press. p. 121 - 159
- Westi, Neil; Eshraghian, Kamran. 1985. Principles of CMOS Design - A System Perspective. Addison-Wesley. p. 310- 333
- Woudsma, Rob; Noteboom, Jos. The Modular Design of Clock Generation Circuits in a CMOS Building System. Journal of Solid State Circuits. IEEE. Vol. SC-20 No. 3. June, 1985. p. 770-774.

APPENDIX A

C PROGRAM FOR ALGORITHMIC PARAMETERS

```

/*****
*   This program determines the parameters, m and k, for the      *
*   cyclical DpS algorithm of Chapter IV. Simple modifications  *
*   will convert this program to solve for m and k for the flow  *
*   mode.                                                         *
*****/
#include <stdio.h>
#include <math.h>

                /* Max available length of shift register chain */
#define MAXSRLEN 64

                /* Expected maximum clock rate for shiftregister */
#define CLK      8e-9

                /* Expected maximum clock frequency */
#define FCLK    1.0/CLK

                /* Minimum Points Per Cycle */
#define MPPC    3

                /* Epsilon to round correctly at extreme values */
#define EPS1    1e-12

```

```
/*
*****
*   Main variable declarations.
*****
main(argc, argv)
    int    argc;
    char   *argv[];

{
    /* Input string holder. */
char    s1[81];

    /* Iteration index. */
int     i;

    /* ASCII to float conversion. */
double  atof();

    /* Desired target frequency. */
double  ft;

double  alpha, Fout, f, j, k, kk, m, error, t, tt;
}
```

```

/*****
*   Read from user the frequency target                               *
*****/
if (argc == 1) {
    printf("\nEnter the frequency to be generated.\n");
    scanf("%f", &ft);
} else ft = atof(argv[1]);

printf("\nThe freq. to be generated is %0.5e.\n", ft);

/*****
*   Initialize                                                         *
*****/
k = 1.0;
tt = 1.0 / ft;          /* Target period */
error = 1.0;
alpha = (int) ((tt * FCLK) + EPS1);

```

```

/*****
*   Determin if frequency target is within range.           *
*   If it is select m and k, and calculate error.           *
*   *****/
if (alpha < MPPC) {
    printf("Frequency out of range.\n\n");
} else {
    if (alpha <= MAXSRLEN) {
        m = alpha;
        k = 1;
        error = tt * FCLK - alpha;
    } else {
        if (alpha >= FCLK) {
            m = 64;
            k = (int) (alpha / 64);
        } else {
            for (i = MAXSRLEN; i >= MPPC; i--) {
                j = (alpha / i) - (int) (alpha / i);
                if (j < error) {
                    kk = (int) (alpha / i);
                    if (kk > 0) {
                        error = j;
                        m = i;
                        k = kk;
                    }
                }
            }
        }
    }
}
ff = (m * k * ft);
f = (int) ((ff + 5E3) / 1E4) * 1E4;
t = 1.0 / f;

Fout = 1 / (m * k * t);

```

```

/*****
*   Print results   *
*****/

printf("The value of alpha is %7.9g.\n", alpha);
printf("The value of t is %7.9g.\n", t);
printf("The value of f is %7.9g.\n", f);
printf("The value of m is %0.0f, of k is %0.0f.\n", m, k);
printf("The frequency generated using these ");
printf("values of m and k is:\n");
printf("Fout = %0.5e.\n", (1 / (m * k * t)));
printf("The value of tt * FCLK - (m*k) ");
printf("(error) is %1.3g.\n", error);

if (ft != 0) {
    printf("The percentage error 100 * (Fout - Ft) ");
    printf("/ Ft is %1.1e%%.\n", 100 * (Fout - ft) / ft);
}

if ((Fout - ft) != 0) {
    alpha = (double) (ft / (Fout - ft));
    if (alpha < 0) alpha *= -1.0;
    printf("The frequency accuracy for these");
    printf("values is %8.8g db.\n", 20.0 * log10(alpha));
}
}
printf("\n\n\n");
}

```

APPENDIX B

BLOCK DIAGRAMS

Digital Arbitrary Waveform Generator

