A High-Resolution Time Interpolator Based on a Delay Locked Loop and an *RC* Delay Line

Manuel Mota, *Member, IEEE*, and Jorgen Christiansen, *Member, IEEE*

***Abstract*—An architecture for a time interpolation circuit with an rms error of ~25 ps has been developed in a 0.7-μm CMOS technology. It is based on a delay locked loop (DLL) driven by a 160-MHz reference clock and a passive *RC* delay line controlled by an autocalibration circuit.**

Start-up calibration of the *RC* delay line is performed using code density tests (CDT). The very small temperature/voltage dependence of *R* and *C* parameters and the self-calibrating DLL results in a low-power, high-resolution time interpolation circuit in a standard digital CMOS technology.

***Index Terms*— Code density test, delay locked loop, high-resolution time measurement, RC delay line, time calibration, time to digital converter.**

I. Introduction

HIGH-RESOLUTION time interval measurement circuits are essential elements of high-energy physics experiments. A typical experiment requires several hundred thousand data acquisition channels. They also find application in other fields, such as range finding, etc. Traditionally, high resolution has been achieved using slow time to amplitude converters, followed by analog-to-digital converters (ADC's) [1]. Such an architecture does not easily satisfy the growing requirements for low power consumption and high integration levels. In addition it relies on technologies with good analog performance. Delay locked loop (DLL) based architectures have been developed to fulfill the same requirements in standard digital CMOS technologies [2]. DLL's are self-calibrating based on a simple external frequency reference (crystal oscillator) making them very attractive for many applications.

The resolution achievable with simple DLL's is, however, limited by the minimum gate delay available in a given technology. Using very fast technologies can improve time resolution at increased cost and power consumption but will in many cases still not be sufficient. A method has previously been proposed by the authors to implement a high-resolution time to digital converter (TDC) using an array of DLL's [3]. This method has been proven to give good time resolution at the expense of increased power consumption because of the additional DLL's.

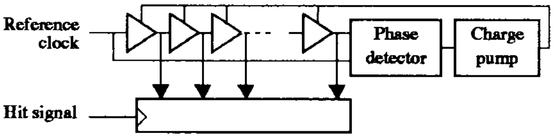

Fig. 1. DLL block diagram.

We describe a novel architecture for a TDC with improved time resolution and reduced power consumption, and we report on the performance of a demonstrator of the architecture for use in high-energy physics experiments.

II. Time Interpolator Architecture

The basis of the proposed interpolator is a single DLL where the dynamic range can be expanded simply by including a clock synchronous counter [4].

In a DLL (see Fig. 1) the reference clock is propagated through a voltage controlled delay line. The delayed signal at the end of the delay line is compared with the reference input. If a delay different from one clock period is detected, the closed loop will automatically correct it by changing the time constant of the delay cells via a charge pump and filter capacitor. An initialization state machine prevents the DLL from false locking at startup.

At the arrival of a hit signal, the status of the DLL is captured in a register. The word stored reflects the arrival time of the hit in relation to the reference clock, resulting in a "time-stamp" measurement [5], [6]. In this simple single N-element DLL scheme the resolution is limited by the basic delay cell ($T_n = T/N$, where T is the reference clock period).

To improve the time resolution of the single DLL one can try to further divide the delay of the delay cells by performing phase interpolation using an array of phase shifted DLL's [3], [6] or weighted sums of consecutive cell outputs [7]. Alternatively, as proposed here, one can sample the status of the DLL several times with a small time interval between the samples. By determining in which sample the rising edge of the reference clock comes out of a delay cell one can deduct the arrival time of the hit with a resolution equal to the sample interval. To get a full time coverage the samples must be uniformly spaced over an interval equal to the delay of the delay cells in the DLL. This method is illustrated in Fig. 2, where the delay of the DLL cells (T_n) has been divided into

Manuscript received March 22, 1999; revised June 8, 1999. This work was supported by a grant from the Portuguese Junta Nacional de Investigação Científica e Tecnológica (JNICT) under the Sup-Programa Ciência e Tecnologia do 2° Quadro Comunitário de Apoio.

M. Mota is with the Laboratory for High Energy Physics, 1000 Lisbon, Portugal and with the European Centre for Particle Physics (CERN), 1211 Geneva 23, Switzerland (e-mail: Manuel.Mota@cern.ch).

J. Christiansen is with the European Centre for Particle Physics (CERN), 1211 Geneva 23, Switzerland (e-mail: Jorgen.Christiansen@cern.ch).

Publisher Item Identifier S 0018-9200(99)08234-7.

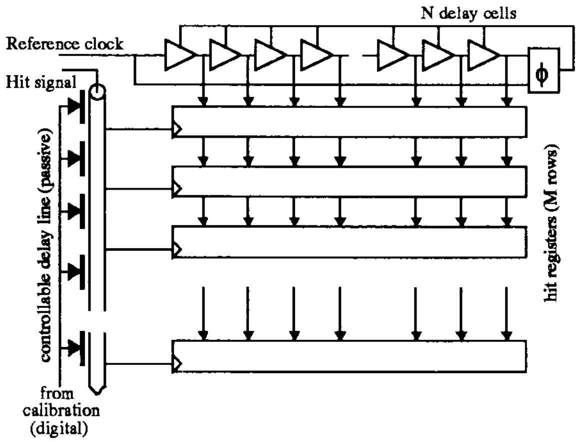

Fig. 2. Timing interpolation circuit.

M subdivisions, resulting in an overall resolution of

$$T_m = \frac{T}{N \cdot M}. \tag{1}$$

Guaranteeing short delays with high precision in an open delay line is not easily done. Active devices (even if they were fast enough) have timing characteristics that vary significantly with supply voltage, operating temperature, and process parameters. Passive *RC* delay lines are, on the other hand, very insensitive to supply and temperature changes. Typical variations are of the order of 500 ppm per Volt or kelvin. However, their delay is to a large extent dependent on IC processing, since parasitic resistance and capacitance parameters are only weakly controlled in a digital CMOS technology. Wide delay variations are thus expected from circuit to circuit, making a calibration of these lines essential to the performance of the proposed architecture.

A. Adjustable RC Delay Line

To be capable of performing a calibration of the *RC* delay line it is necessary make it adjustable. A digital adjustment scheme seems to be the most appropriate because of its simplicity and its immunity to potential noise sources. Of the many possible digital adjustment methods, two will be briefly described.

The first scheme consists of partitioning the delay line into small fixed segments with a sensing buffer at the output of each segment (Fig. 3). The adjustment of the delay taps is performed simply by selecting the best segment buffer to drive each individual tap. A complication of this otherwise simple scheme is that the total variation with process is so large (±30%) that some segment buffers must be capable of being connected to one of several taps requiring a carefully planned layout of the delay line.

An alternative scheme uses a lumped variable capacitor connected to fixed positions along the delay line, as illustrated in Fig. 4. Varying the value of the capacitor changes the time constant of the line and therefore adjusts its effective delay. The variable capacitance is simply implemented as a bank of fixed capacitors which can be switched in or out. The advantage of this scheme is that the configuration of the *RC* delay line itself is fixed and only an additional parasitic capacitance is used to obtain the required adjustment capability. A disadvantage of this scheme is that when one tap is adjusted it also influences the other taps on the delay line. This slightly complicates the autocalibration procedure, as will be explained later.

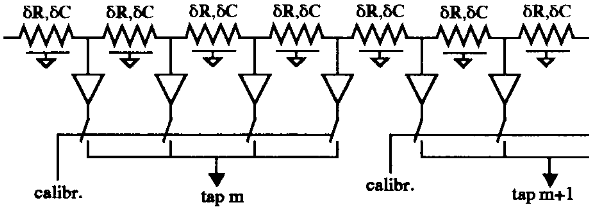

Fig. 3. Adjustment by tap selection.

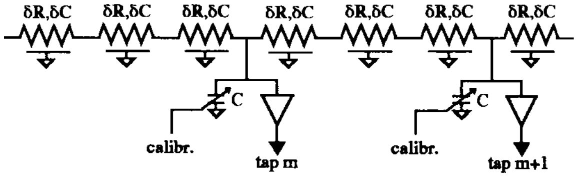

Fig. 4. Adjustment using variable capacitance.

B. Autocalibration

The automatic self-calibration of the time interpolation circuit is a critical part of the architecture to obtain improved time resolution in comparison with alternative implementations. The DLL automatically locks itself to the reference clock and does not need any further calibration (more on this later). The total delay of the *RC* delay line must be matched to the delay cells in the DLL. The individual tap positions must be evenly distributed over the total delay of the *RC* line.

In principle, each chip could be calibrated when production testing is performed by laser trimming or blowing internal fuses. However, this would require that each chip be calibrated to be used with a precisely defined clock reference frequency and would not allow for an optimal calibration taking into account the working environment of the IC (temperature, supply voltage). It is preferable to apply a calibration procedure which can be performed *in situ* and in addition enable the IC to verify its correct calibration (and recalibrate) at regular intervals.

The total delay of the *RC* line could be calibrated using the known absolute delay of one of the delay cells in the DLL. However, this would not enable a precise adjustment of each *RC* delay tap individually. In addition, one would need to choose one delay cell from the DLL as a reference. The delay cells in the DLL have some variation due to mismatch, and choosing one as an absolute reference is not the best solution.

The optimal calibration procedure would use the average delay of the DLL cells as a reference and enable each *RC* tap to be calibrated individually. A statistical code density test offers all these advantages and is also relatively simple to implement completely on chip. In a code density test for a time interpolator a large number of random hits must be generated and the number of hits registered in each time bin histogrammed. The relative difference between the bin contents in the histogram is a direct measure of the relative size of each time bin. The histogramming can be performed for all the individual bins in the circuit ($N * M$ bins) to obtain a detailed characterization of the combination of the DLL and the *RC* delay chain. For characterizing the *RC* delay chain alone, the values for the same *RC* delay tap can be summed across the DLL, thereby obtaining a measure of the *RC* delay averaged over the DLL. The summing can in fact also be performed across the *RC* delay chain thereby giving a measure of the DLL averaged over the *RC* delay chain. In this simple way the characteristics of the DLL and the *RC* delay chain can be characterized individually and be used to perform an autocalibration. The statistical nature of this calibration procedure intrinsically ignores all sources of random noise (clock jitter, DLL jitter, thermal noise, etc.); this must be considered an additional advantage.

The statistical code density test is an accurate way to characterize the performance of the circuit and it only requires a random hit generator and a simple arithmetic unit to derive the optimal calibration parameters from the histogram data. The random hit generator can be a simple uncorrelated oscillator. The arithmetic unit consists only of a few accumulators and comparators. It is very important that the random hit generator runs completely uncorrelated to the reference clock, otherwise beating effects will seriously compromise the basis for the code density test. The calibration procedure can be run in an iterative fashion whereby the accuracy of the calibration can be improved. This is especially the case when the *RC* delay line is implemented with the variable lumped capacitance, where the adjustment of one tap will affect the other taps. A more detailed overview of the calibration algorithms is given in the Appendix.

The statistical nature of the code density test requires a certain number of random hits to be generated to obtain a sufficient confidence level. Assuming a normal approximation of the test statistic, a similar procedure to the one developed in [8] can be used to obtain the required number of hits. It results in the following equation, where $z_\alpha/2$ is the normal variable z that corresponds to an area under the standard normal distribution curve of $\alpha/2$

$$N_{\text{hit}} = \left(\frac{{}^{z}\alpha/2}{\beta}\right)^2 \cdot (M-1). \tag{2}$$

If we accept a tolerance of 10% in the results ($\beta = 0.1$) with a 99% confidence level ($\alpha = 0.01$), then $z_{\alpha/2} = 2.575$ and less than 8192 (2^{13}) hits are required for M equal to 8.

In Fig. 5 the characteristics of the delay line (using the tap selection scheme) before and after calibration are shown. The uncalibrated delay control parameters had previously been extracted from typical case simulations of the *RC* delay line. It can be seen that the simulation model used was quite accurate and only minor improvements had to be obtained from the autocalibration.

III. Demonstrator

A demonstrator circuit was designed in a 0.7 μm CMOS technology (see Fig. 6). This circuit proved the feasibility of the proposed architecture, including the automatic calibration based on a code density test.

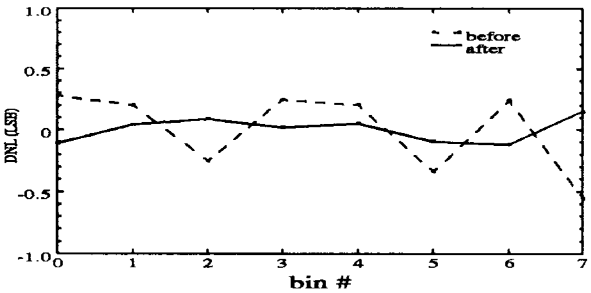

Fig. 5. Delay line differential nonlinearity before and after calibration (measured on the prototype chip).

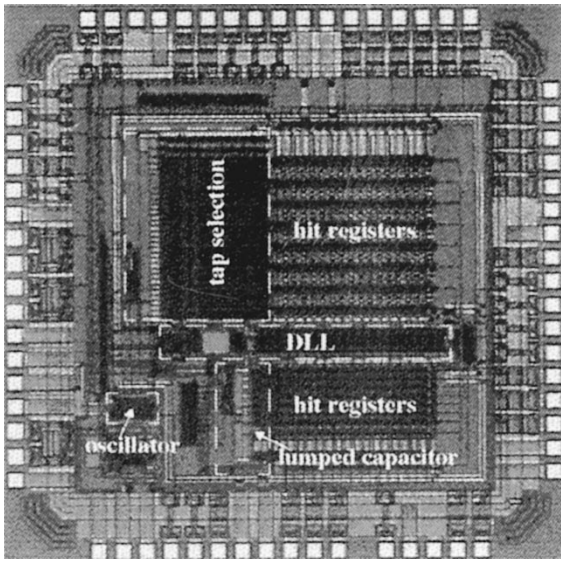

Fig. 6. Prototype circuit showing main functional blocks.

It includes a single DLL, shared between two interpolation channels. One channel uses an adjustable *RC* delay line based on the tap selection scheme and the other uses the lumped capacitor scheme as previously described. A free running oscillator needed for the autocalibration was also implemented. The 160-MHz reference clock is provided by an external crystal oscillator.

The circuit uses 10.7 mm^2 of silicon area and was packaged in a 68-pin ceramic JLCC.

A. The DLL

The DLL building blocks were taken from a previous design as these have already been extensively studied and proven to perform well [3]. Using the same basic elements also makes it possible to compare two different architectures based solely on their time interpolation merits. The DLL delay cell is made of two current starved inverters, as shown in Fig. 7. The delay control range is split into smaller ranges that can be digitally programmed via the signals sel_rangeN and sel_rangeP. This way the delay cell can be forced to operate in the range that results in a better delay matching and less jitter, regardless of the operating conditions.

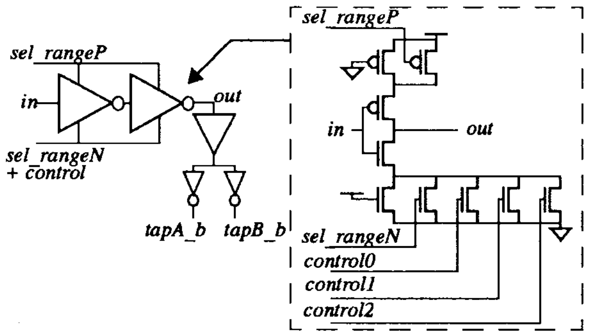

Fig. 7. DLL's delay cell (from [3]).

A two-state phase detector based on a balanced D flip-flop [9], a charge-pump, and an integrating capacitor complete the control loop. The control voltage that it generates is fed to the delay cells via the *control0..2* signals.

The contribution of the DLL to the interpolation nonlinearity should be kept small. Main contributors for its nonlinearities are device mismatch [10], edge effects, and clock correlated noise. Power supply grid separation, the use of dummy cells at the extremities of the delay line, and matching-minded device sizing can effectively reduce cell delay variations to the order of 1%. The DLL has 16 delay elements giving a delay unit $T_n = 390$ ps. The jitter in the longer DLL's of previous chips [3] has been measured to be below 18 ps rms.

B. Calibration Oscillator

Random hits are obtained from a simple three-inverter free running oscillator, whose frequency is reduced to ~10 MHz by the use of an asynchronous counter. The oscillator can be stopped, after calibration has been performed, to conserve power. The oscillation frequency has been made programmable to be capable of evaluating the calibration procedure with different settings and possible beating problems with the reference clock.

Tests performed using various oscillation frequencies have shown that, for particular oscillator settings, beating would occur. It is, however, straightforward to avoid the beat frequency.

C. RC Delay Line

The *RC* delay lines were constructed with segments of polysilicon over thick-oxide. In the given technology the length and width of a typical segment are 30 and 40 μm, respectively. Other options, such as using polysilicon over thin-oxide, were discarded because the small time constant required would result in segments with very awkward aspect ratios.

A detailed Spice model was developed in order to obtain accurate estimates of the characteristics of the delay line. A multisection T network, with an appropriate number of sections, is required to simulate the distributed *RC* delay line with sufficient precision (Fig. 8). Simulation accuracy may be traded for speed depending on the number of T-elements into which the line is divided. A network including lumped loads, resistance of contacts, active buffers, and the distributed *RC* line was then built from this.

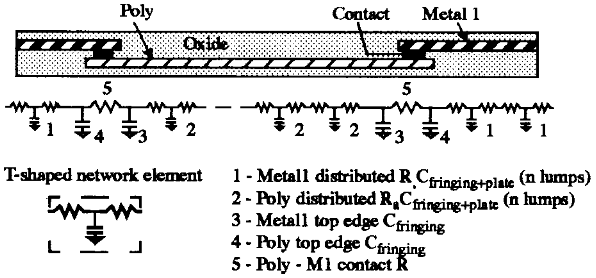
Poly
Contact
Metal 1
Oxide
5
5
1
4
3
2
2
2
3
4
1
1
T-shaped network element
1 - Metal1 distributed R,C fringing+plate (n lumps)
2 - Poly distributed R$_s$C fringing+plate (n lumps)
3 - Metal1 top edge C fringing
4 - Poly top edge C fringing
5 - Poly - M1 contact R

Fig. 8. Simulation model of segment of *RC* delay line.

The total delay from the hit input until the hit registers latch the status of the DLL also includes delays in active buffers and the hit registers themselves. As all these active elements are identical circuits and the architecture is only sensitive to delay differences, their delay, in principle, cancels out. Mismatching between these elements will, however, be visible, but it can to a large extent be compensated for by the autocalibration.

The design goal is to subdivide the DLL delay cells ($T_n =$ 390 ps) into eight fine time bins requiring a 390 ps/8 = 48.8 ps delay per segment of the *RC* delay line.

D. Performance Analysis

In the given configuration with a fine time bin size [herein referred to as the least significant bit (LSB)] T_m = 48.8 ps, the theoretical rms resolution is given by

$$\sigma_{\text{bin}} = \frac{T_m}{\sqrt{12}} = 14.1 \text{ ps.} \tag{3}$$

Matching limitations in the DLL degrade the interpolator resolution. The maximum cumulative effect of cell mismatch, σ_{dll}, is seen in the middle of the DLL [11]. Assuming a matching (σ_{match}) of 1%, an additional rms error of the DLL (σ_{dll}) is

$$\sigma_{\text{dll}} = \sigma_{\text{match}} \cdot \frac{\sqrt{N}}{2} \cdot T_n = 7.8 \text{ ps.} \tag{4}$$

Delay line calibration acts on the integral nonlinearity, limiting it to a maximum of ±0.5 LSB. Therefore the nonlinearity of the passive delay line accounts for a maximum σ_{dl} of

$$\sigma_{\text{dl}} \cong \frac{T_{m \cdot 0.5}}{\sqrt{12}} = 7.1 \text{ ps.} \tag{5}$$

The closed loop of the DLL has some intrinsic jitter which has been simulated and measured to be of the order of 10 ps.

Adding these contributions quadratically, the estimated overall rms resolution should be ~20.2 ps (~0.41 LSB).

IV. Test Results

Characterization of the differential and integral nonlinearities (DNL and INL) was performed using statistical code density tests from a data set of 80 000 random hits (using both an external and the internal hit oscillator). Separate tests were performed on the two different channels implemented. No significant difference was found between using an external hit generator or the on-chip generator, proving that a sufficiently uncorrelated oscillator can be made on-chip.

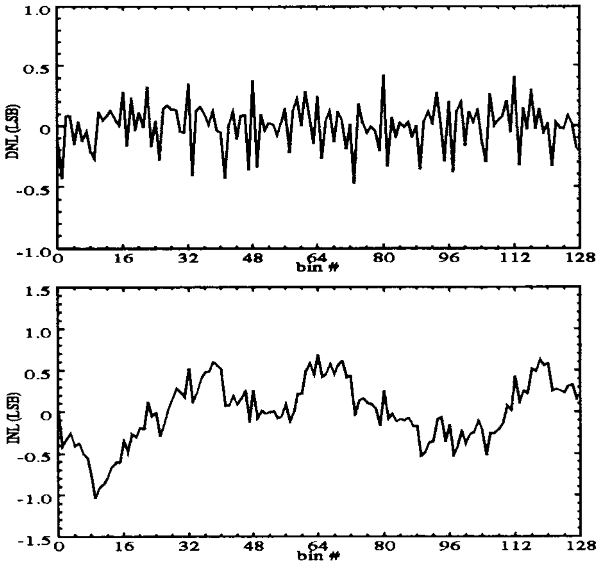

Fig. 9. DNL and INL using programmable tap selection.

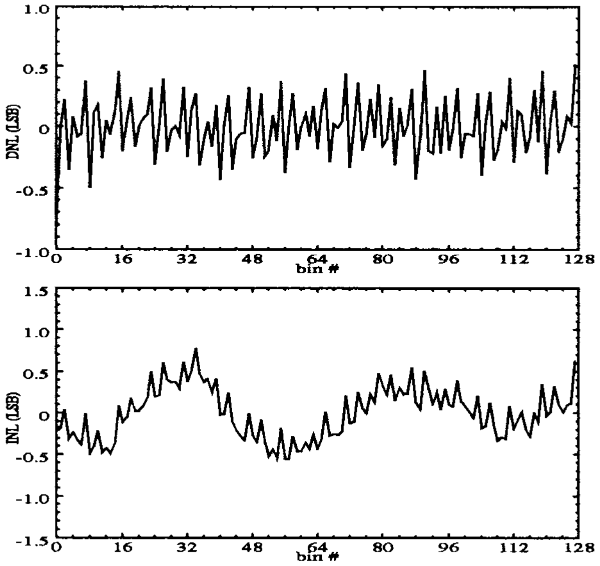

Fig. 10. DNL and INL histograms using lumped capacitance.

TABLE I
MEASURED LINEARITY AND CONVERSION ERROR

Channel	Tap selection	Lumped capacitance
LSB (ps)	48.8	48.8
DNL (σ/max) (LSB)	0.18/0.47	0.24/0.79
INL (σ/max) (LSB)	0.37/1.04	0.30/0.78
error (σ/max) (LSB)	0.51/1.60	0.44/1.55

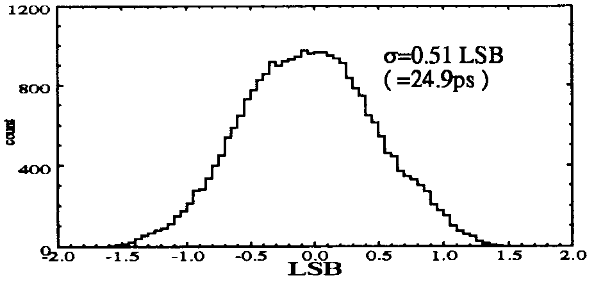

Fig. 11. Conversion error histogram using tap selection.

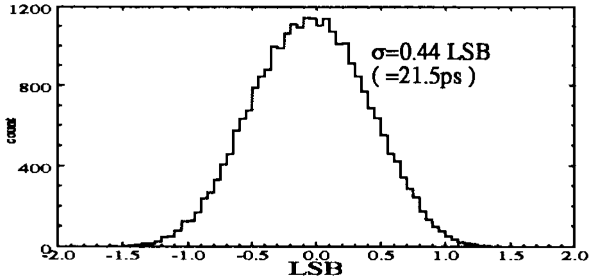

Fig. 12. Conversion error histogram using lumped capacitance.

The resulting histograms, obtained after calibration of the delay line (at room temperature), are shown in Figs. 9 and 10 and summarized in Table I.

A linear time sweep over the complete dynamic range was performed in order to measure errors of random nature such as jitter, internal and external noise, etc. These kinds of errors are not captured by statistical tests due to their random behavior. A motor-driven coaxial phase shifter was used to generate a time sweep. This passive instrument is very linear and generates very little jitter, making it better suited for this kind of test than active delay generators. The plots in Figs. 11 and 12 show the conversion error histograms for the two channels. They include 26 400 measurements each, generated using a time step of ~2.4 ps (accumulating ten measures per time step).

An important feature of the proposed architecture is its potential insensitivity to temperature changes. It has been tested over temperature changes of 30 °C keeping the same calibration parameters. The measured rms resolution was found to degrade less than 5% (~1 ps) and the delay of the passive *RC* line to vary less than 2.5%. This minor degradation can, if required, be recovered by recalibrating after a significant temperature change.

The power consumption of the demonstrator chip was measured to be 220 mW with a 5-V supply voltage. Most of this is spent in the DLL. This represents a significant reduction from 28 W [1], 800 mW [3], and 5.7 W [7] reported in circuits whose performances are summarized in Table II.

A. Error Sources

The nonlinearities of the DLL were found to be the dominant error source in the implemented circuit. In the proposed architecture the matching of the delay cells must be better than a few percent to prevent it from becoming the dominant source of error. This is illustrated in Fig. 13 where the INL of the DLL and the INL of the complete interpolation circuit are shown together. It is clearly seen that the integral error of the whole circuit closely matches the integral error of the DLL alone.

TABLE II
SUMMARY OF DIGITIZER PERFORMANCES

Ref.	technology	LSB (ps)	σ (ps / LSB)	Power (W/channel)
present	0.7μm CMOS	48.8	21.5 / 0.44	0.11 (2 channels)
[3]	0.7μm CMOS	89.3	33.9 / 0.38	0.2 (4 channels)
[7]	Bipolar f_t= 25GHz	15.6	not reported	5.7 (1 channel)
[1]	discrete components (ECL)	29	15.1 / 0.52	1.75 (16 channels)

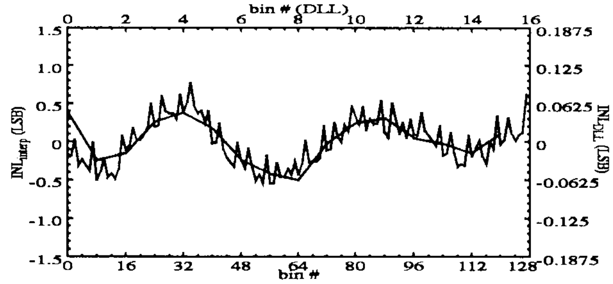

Fig. 13. INL of DLL and complete time interpolator circuit.

The matching in the implemented DLL was measured to be of the order of 3–4% (rms), slightly worse than expected. Significant contributions for the cell mismatch are estimated to be clock related supply noise and device mismatch.

V. IMPROVING PERFORMANCE OF PROPOSED ARCHITECTURE

In the implemented demonstrator circuit it was clearly shown that the matching properties of the DLL were the major factors limiting time resolution. However, this problem can be solved relatively easily by using the described autocalibration procedure to adjust the individual delay cells of the DLL as well. This adjustment can be performed by programmable correction of each delay element itself or by including adjustable passive delays on the outputs of the delay taps from the DLL, as illustrated in Fig. 14. This should in principle work well, as matching is a static effect having little dependency on working conditions. A differential implementation of the DLL should also improve its noise sensitivity.

It is evident that a significant performance improvement can be obtained using a more modern submicron technology. It is estimated that switching from the 0.7-μm technology to a 0.25-μm technology would bring a factor four reduction to the delay of the delay cells used in the DLL. Reducing the delays in the *RC* delay line should in principle also be trivial. In submicron technologies the matching properties are often relatively poor, but this can be handled by using the proposed autocalibration procedure for both the DLL and the *RC* delay line. Obtaining a time resolution better than 10 ps will, however, be very difficult as jitter in the DLL and thermal noise will start to dominate.

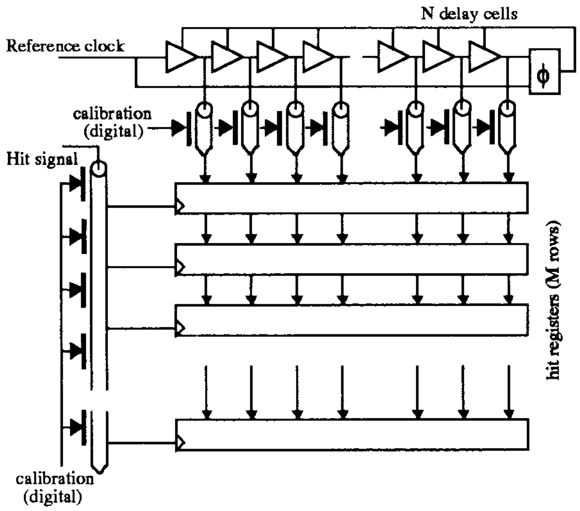

Fig. 14. Improved timing interpolation circuit.

VI. CONCLUSION

A high-resolution time interpolator prototype IC was produced to validate the proposed autocalibrating architecture. The performance of the prototype shows that it is possible to obtain an rms resolution better than 25 ps in a low-power monolithic circuit using a digital 0.7-μm CMOS technology. The proposed architecture has the potential of achieving significantly improved resolution in more modern submicron CMOS technologies.

APPENDIX
DELAY LINE CALIBRATION ALGORITHMS

In the case of the *RC* delay line implemented with a tap selection scheme, the calibration algorithm is quite simple, since each tap can be calibrated individually. It is sufficient to select, for each tap, the correct line segment output that leads to the minimum nonlinearity.

In the case of the *RC* delay line implemented with a variable lumped capacitor scheme, the calibration algorithm is made more complex by the way the adjustment of each tap influences the others.

The calibration algorithm profits from two properties inherent to this scheme: a simultaneous unit capacitor change in all the banks results in a uniform variation of the delay of all taps. On the other hand, the change of a unit capacitor in a single bank will only have a small influence on the delay of the taps that are located toward the beginning of the line. This influence decreases as the taps are located further away from the bank that was changed.

It is therefore possible to approach the correct calibration parameters by uniform, but coarse, steps by varying all the capacitor banks simultaneously. The coarse step starts with the smallest line delay and ends when the delay of the line is within a given interval of (but smaller than) the ideal delay.

Detailed calibration can then be performed by small iterative adjustments to the individual taps one by one. Since the calibration of all taps is approached from shorter delays, the small cross influence will not be detrimental for the final calibration.

These calibration algorithms are intended to reduce the nonlinearity of the *RC* delay line to an acceptable value. Their convergence is determined by the width of the nonlinearity acceptance interval. However, if this interval is wider than the minimum discrete adjustment step available, their convergence is guaranteed.

References

[1] O. Sasaki, T. Taniguchi, T. K. Ohska, and H. Kurashige, "A high resolution TDC in TKO BOX system," *IEEE Trans. Nucl. Sci.,* vol. 35, Feb. 1988.

[2] T. E. Rahkonen and J. T. Kostamovaara, "The use of stabilized CMOS delay lines for the digitization of short time intervals," *IEEE J. Solid-State Circuits*, vol. 28, pp. 887–894, Aug. 1993.

[3] M. Mota and J. Christiansen, "A four channel, self-calibrating, high resolution, time-to-digital converter," in *Proc. IEEE ICECS'98,* vol. 1, pp. 409–412.

[4] C. Ljuslin, J. Christiansen, A. Marchioro, and O. Klingsheim, "An integrated 16 channel CMOS time-to-digital converter," *IEEE Trans. Nucl. Sci.,* vol. 41, pp. 1104–1108, Aug. 1994.

[5] Y. Arai and M. Ikeno, "A time digitizer CMOS gate-array with 250 ps time resolution," *IEEE J. Solid-State Circuits*, vol. 31, pp. 212–220, Feb. 1996.

[6] J. Christiansen, "An integrated high resolution CMOS timing generator based on an array of delay locked loops," *IEEE J. Solid-State Circuits,* vol. 31, pp. 952–957, July 1996.

[7] T. A. Knotts, D. Chu, and J. Sommer, "A 500 MHz time digitizer IC with 15.625 ps resolution," in *Proc. IEEE ISSCC'94*, vol. 37, pp. 58–59.

[8] J. Doernberg, H.-S. Lee, and D. A. Hodges, "Full speed testing of A/D converters," *IEEE J. Solid-State Circuits*, vol. 19, pp. 820–827, Dec. 1984.

[9] M. G. Johnson and E. L. Hudson, "A variable delay line for CPU co-processor synchronization," *IEEE J. Solid-State Circuits*, vol. 23, pp. 1218–1223, Oct. 1988.

[10] M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching proprieties of MOS transistors," *IEEE J. Solid-State Circuits,* vol. 24, pp. 1429–1433, Oct. 1989.

[11] S. Kuboki, K. Kato, N. Miyakawa, and K. Matsubara, "Nonlinearity analysis of resistor string A/D converters," *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 383–390, June 1982.

Manuel Mota (S'91–M'91) was born in 1968 in Coimbra, Portugal. He received the degree in electrical engineering from the University of Coimbra in 1991. He is currently pursuing the Ph.D. degree at the Technical University of Lisbon, Portugal.

From 1992 to 1995, he was with the Laboratory for High Energy Physics (LIP), Lisbon, designing digital ASIC's for high-energy physics experiments. Since 1996, he has been at the European Centre for Particle Physics (CERN), Geneva, Switzerland. His current research interests include digital and mixed-mode integrated circuits and high-resolution time to digital converters.

Jorgen Christiansen (M'90) was born in 1961 in Lyngby, Denmark. He received the M.S. degree from the Danish Technical University, Copenhagen, in 1986.

From 1986 to 1989, he worked for Scantest Systems, Denmark, designing special-purpose testers for integrated circuits. Since 1989, he has been working for the European Centre for Particle Physics (CERN), Geneva, Switzerland, designing special-purpose integrated circuits for high-energy physics experiments. His current interests include mixed-signal integrated circuits, high-resolution time to digital converters, and high-speed serial links.