ITU-Compliant Macrocells for Dual Tone Multiple Frequency Transmission and Reception

Arturo Purroy
Universidad de Zaragoza, Centro Politécnico Superior
c/María de Luna 3, 50018 Zaragoza (Spain)
Email: apurroy@unizar.es

Isidro Urriza
Universidad de Zaragoza, Centro Politécnico Superior
c/María de Luna 3, 50018 Zaragoza (Spain)
Email: urriza@unizar.es

***Abstract*— New applications for domotic environments demand the use of the telephone network to provide access points. Due to the inherent limitations of conventional telephone lines, dual-tone multiple frequency (DMTF) signaling is still in use. Previously published solutions are targeted to commercial devices such as digital signal processors (DSP) or microcontrollers. We present an efficient VHDL implementation of a transmitter and a receiver for DTMF signaling in two macrocells. Our architecture includes efficient filtering schemes, improved sequential mathematical operators, and robust power detection. The receiver is based on a low-complexity algorithm in order to reduce the size of both macrocells.**

I. Introduction

The telephone network provide fast access to almost anywhere in any industrialized country. Being so wide added to the fact that using this network is cheap, makes it perfect for controlling domotic environments. An easy interface must be offered to customers in order to manage the home automation system, and the phone push-button keyboard may be an efficient solution.

Inside the house, a controller device is needed. This device should be able to decode the DTMF symbols received from the user and link with the home systems to perform the asked service. To develop cheap solutions, single chip designs are preferred. We present in this paper the design of a digital transmitter and receiver of DTMF symbols modeled in a hardware description language (VHDL), which can be easily added to other designs in a single field programmable gate array (FPGA). The efficiency and physical size of the final design have been the main goal of the research.

Several chips are available which employ analogic circuitry to generate and decode DTMF signals. The advantages of a digital system include better accuracy, precision, stability, versatility, and reprogrammability as well as lower chip count, and thereby reduced board-space requirements.

Most of previous DTMF generation methods prefer the Direct Digital Synthesis (DDS) technique [1] in order to generate all the needed frequencies [2], although there are some that use digital sinusoidal oscillators to implement tone generation [3]. However, digital oscillators require mathematical complex operators (multipliers) and don't offer enough versatility to generate the eight DTMF frequencies efficiently.

A DDS generator can be implemented by an accumulator and a small look-up table, changing the output frequency with the constant which is added each cycle. The simplicity and low hardware cost of this solution makes it fit perfectly in a DTMF generator.

There have been several investigations dealing with efficient DTMF detectors, trying to simplify the standard filter bank. DTMF detectors typically consist of a signal analysis front end followed by a decision logic back end. The usual approach is to use an adaptable filtering architecture which is able to implement multiple filters with just the hardware of one.

The Goertzel algorithm [4] is the basis of some DTMF detectors [3]. It is supposed to be much faster than a true fast Fourier transform (FFT), as only a few of the set of spectral line values are needed and only for those values are filters provided. The use of two-pole IIR filters effectively computes values of the discrete Fourier transform (DFT), but for each input value many products must be computed.

Using two sliding windows, some algorithms based on the nonuniform discrete Fourier transform (NDFT) [5] meet all of the ITU recommendations. The DTMF detector applies frequency detection to windows of samples, a longer window is needed for the low-group frequencies as the bandwith is narrower. These windows have amplitude values which not always are integer powers of two, what makes the implementation more difficult.

Different frequency estimation techniques have been applied, although most of them are expensive to implement. Subspace techniques such as Multiple Signal Classification (MUSIC) and adaptive techniques such as Least Mean-Square (LMS) estimation have been tested in real-time implementations. However, the results have not been completely satisfactory [6].

A very efficient solution was proposed by A.A. Deosthali, S.R. McCaslin and B.L. Evans [7]. Using adaptive notch filters and sophisticated decision logic, their detector meets the ITU standard when implemented in a 8-bit microcontroller. Based in this algorithm, our design has rearranged all the filters and signal processing units to take advantage of a specific hardware design. As the original implementation used a 8-bit microcontroller, when more precision was needed (16-bit data), the original algorithm had difficulties to operate. This and other implementation problems have been solved in our design.

Our design is the first solution published that provides

VHDL code encapsulated in digital macrocells. This goal has added constraints to our design, as the size of a macrocell should be minimized. The size of the macrocell which implements the transmitter is less than 1,000 equivalent gates, while the receiver uses about 10,000 equivalent gates.

II. The DTMF signaling

Each button in a phone keyboard is coded into a DTMF symbol and sent through the telephone network. Each symbol is transmitted as two tones of the frequencies that are shown in Table I. The exact frequencies have been chosen to reduce the presence of harmonics, but this decision increases the complexity of any implementation, as the frequencies are close and have arbitrary values.

TABLE I
Frequencies of DTMF symbols

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	#	0	*	D

Both the transmitter and receiver of DTMF symbols must be very versatile to generate the eight frequencies and detect them efficiently. The International Telecommunication Union (ITU) DTMF standard not only specifies the frequencies of each DTMF symbol, but it also states time duration, interference levels and other symbol characteristics [8] which the receiver must comply, as it can be seen in Table II. As the macrocells we have designed will be encapsulated in bigger designs, the specifications regarding power levels are irrelevant.

The receiver is the most complex part, as the signal processing required to decide whether the input signal is one of the sixteen symbols, and which one it is, has a high mathematical load. High order filters are needed for the standard frequencies and tolerances, what makes the filter bank solution too hardware demanding.

III. Transmitter Design

The transmitter is based on the DDS technique. In its basic configuration, a phase accumulator with a fixed step drives a memory in which the values of the sine amplitude are stored (see Fig. 1). Changing the constant F added each cycle, the output frequency can be easily controlled.

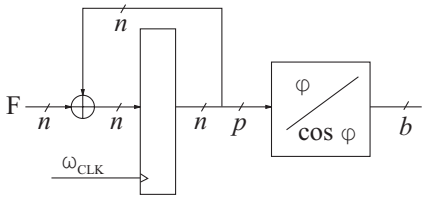

Fig. 1. Generic DDS scheme

The best solution is to implement two DDS architectures, one for each frequency group of the DTMF recommendation. The parameters n, p and b take the lowest values which make the transmitter comply with the standard. The value of n sets the frequency resolution $\Delta\omega$ given by equation (1), and only p bits of the phase are used to drive the memory, what will generate phase noise. The number of bits used in the amplitude of the sine are related to the quantization noise.

Through extensive simulation we have obtained the optimum values for a standard DDS scheme: $n = 8$ bits, $p = 4$ bits, and $b = 3$ bits. An extra bit for n is needed to achieve the required frequency resolution of the low frequency group, as it is dependant on a given percentage (1.5 %) of the lowest frequency of each group.

$$\Delta f = \frac{f_s}{2^n} \leq f_{min} \cdot 0.015 \qquad (1)$$

where f_s is the sampling frequency and f_{min} the lowest frequency of the group.

These optimum values comply with the ITU recommendation implemented in a standard DDS scheme. However, the hardware implemented can be reduced by taking advantage of the symmetry of the sine waveform. As the final parameters are low enough, we have also avoided using a memory to store the sine values. Moreover, wiring the amplitude values to their logic value, both DDS effectively use the same hardware.

The two frequencies of each symbol are generated by two accumulators of 8 and 9 bits, and eight constants of 3 bits representing the amplitude of the sine waveform. These numbers give an idea of the high effort made to reduce the hardware of the macrocells. The adders needed to implement both accumulators are ripple-carry adders (RCA), which require the smallest area of silicon. Their slowness is not a problem, as the low sampling frequency allows long combinational paths.

There is still another constraint to meet, as the timing specifications demand the use of a finite state machine. For 68 milliseconds, the transmitter sends the sum of the two tones, giving more gain to the higher frequency tone, hence setting the twist to 3.5 dB. After each symbol is sent, the transmitter sends a silence for the same duration of the symbol, ignoring any other petition from the keyboard until the whole symbol (the two tones and the silence) have been sent.

In Fig. 2, the frequency spectrum of a DTMF symbol is shown. Both the SNR recommendation and the twist recommendation are met.

The output of the macrocell is the DTMF symbol sampled at 8 kHz and represented in 2's complement.

IV. Receiver Design

The architecture of the receiver is based on a sequencer, controlling all the processing blocks. Each block implements several mathematical operations multiplexing in time a very basic hardware. In addition, the blocks of the original algorithm [7] have been grouped, based on the similarity of their mathematical structure, and are executed by the same hardware. Data is stored in 8-bit registers, except in the period estimators, where 16-bit registers are used.

TABLE II
ITU SPECIFICATIONS FOR DTMF SYMBOLS

Frequency tolerance	Valid	$\leq \pm 1.5\,\%$ of above frequencies
	Invalid	$\geq \pm 3.5\,\%$ of above frequencies
Signal duration	Valid	40 ms minimum
	Invalid	23 ms maximum
Signal exceptions	Pause duration	40 ms maximum
	Signal interruption	10 ms minimum
Twist	Forward	8 dB
	Reverse	4 dB
Signal strength	SNR	15 dB minimum

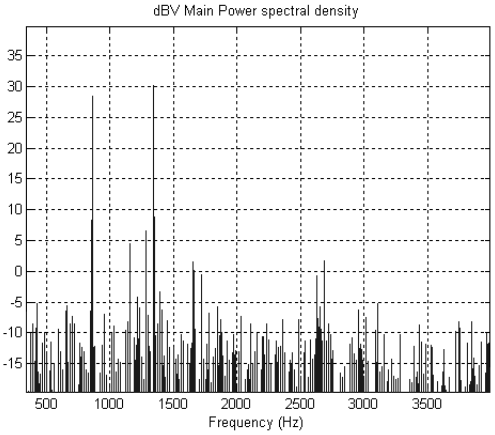

Fig. 2. SNR of a transmitted symbol

In Fig. 3 it is shown an approximate scheme of the different blocks with which the receiver is built. Each of these blocks represents a single hardware structure which executes multiple signal-processing operations (e.g. filters, estimators, etc.).

The input signal is decimated by two using a two-tap moving average antialiasing filter, what halves the sample rate. Once decimated, the signal is filtered by a notch filter centered at 300 Hz, which eliminates the dial tone of the telephone line. After this filter is applied, the reception path splits in two to detect the higher frequency of the symbol and the lower one. Two notch adaptive filters suppress the frequency detected in the other path, the main interference for the detector.

The three notch filters just described are FIR filters and share the same hardware. Hence, we have designed an efficient architecture (see Fig. 4), using just one 8-bit adder and an internal register to perform all the mathematical operations that the three notch filters require. Adapting a sequential signed multiplier to our needs and using advanced shifting techniques to reduce the number of multiplexors, the size of the final hardware is very small.

The frequency estimators use a highly accurate algorithm [9] based on zero crossings. Before counting the cycles between two zero-crossings, the DC component must be eliminated. If the signal has any DC component, the estimation will have an important bias. The mutliplicative constants of

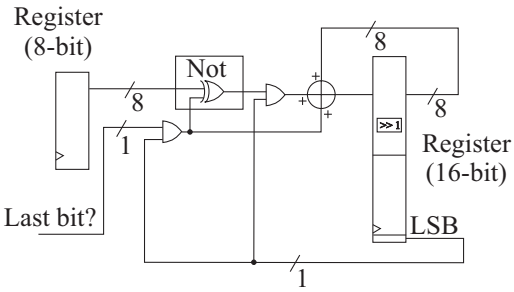

Fig. 4. Multi-filtering architecture

the DC filters are integer powers of two, what is implemented without any extra hardware, just wiring the bits shifted. All the partial results are summed up with the adder used to implement the next section of the algorithm.

As the possible DTMF frequencies are close between each other, the number of cycles in a period is the same for different frequencies. In order to obtain an accurate estimation of the period, the algorithm calculates the fraction of cycle that completes the period. A division is required, and this mathematical operator has been implemented in an efficient sequential architecture (see Fig. 5) which can also be used to perform other operations.

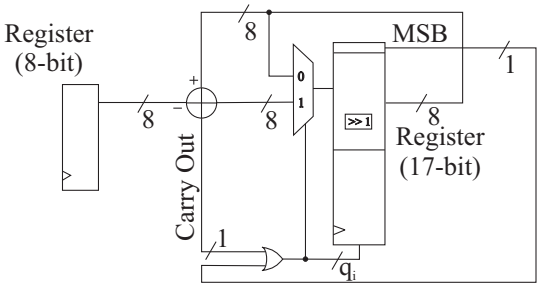

Fig. 5. Divisor/Accumulator architecture

The integer and fractional parts of the period are calculated and low-pass filtered to eliminate spurious values. In the original algorithm, both parts are stored in 8-bit registers, what implies the use of additional hardware to adjust the results of the filters. The overflow of the fractional part must be added to

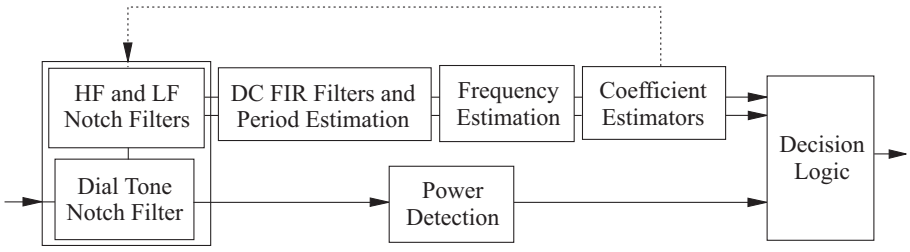

Fig. 3. Block diagram of the receiver

the integer part, and the relation also applies in the other way. We have used 16-bit logic to perform all the operations that involve the integer and fractional parts, reducing considerably the amount of hardware required.

In order to decide which frequency is being received we use a binary tree comparator. Each frequency is assigned an integer part and an interval of fractional values which were chosen according to the DTMF recommendations. If the value obtained from the current input sample does not fit any correct interval, the decision logic should decide whether the symbol has ended or the pause has been produced by noise.

The other input of the decision logic is the output of the power detector. The robust estimation generated with the hardware in Fig. 6 is compared with an adaptive noise floor. Instead of using the absolute value of the input and multiplying it by 2 to estimate the power, what requires an adder to be calculated, we use a simplified estimator. When the input value is less than zero, we negate that sample, we shift the value to multiply it by 2 but inserting a logic 1 instead of a logic 0, what reduces the overall error in the estimation. This method replaces an 8-bit adder with eight *XOR* gates.

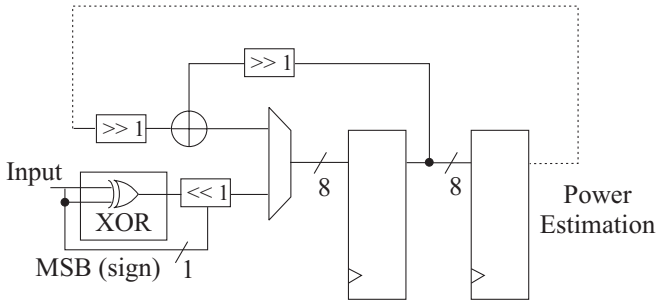

Fig. 6. Power estimator architecture

All the timing recommendations are verified by the decision logic. We have split in two parts the combinational logic in order to decrease the length of the combinational paths. This last block updates the output symbol and rises a 1-bit output signal each time a new symbol is detected.

Both sequential multiplications and sequential divisions used in the proposed hardware require as many cycles as bits has the data they operate with. As all the operations and filters are executed sequentially, for each 8 kHz cycle our system needs 2^{10} cycles. With the ten bits of the sequencer, all the stages of the algorithm are easily controlled without a complex control logic.

V. Testing and Implementation

The VHDL model has been extensively simulated in order to verify its compliance with the DTMF recommendation.

The receiver has demanded a higher effort to verify it. The minimum duration of each detected symbol has been calculated sending bursts of 10 identical symbols to the receiver, and looking for the duration which made all of them be correctly detected. The simulation of the symbol '5' in Fig. 7 shows a correct detection with a duration of 34 ms.

The pause duration needed to detect two different symbols has been simulated in a similar way. In Fig. 8 it can be seen that the recommendation for the pause duration has been met.

All the recommendations are met by our macrocells. Some of the specifications of the receiver are shown in Table III. Comparing these numbers with those in Table II, we can see that, as the authors of the algorithm claim, the receiver complies with the ITU standard for DTMF signaling.

TABLE III
Specifications of the receiver

Frequency error	Minimum	3.2 %
	Maximum	3.4 %
Signal duration	Minimum	32 ms
	Maximum	35 ms
Twist	Forward	3.5 dB
Signal strength	SNR	15.2 dB

Once both macrocells had been designed and simulated, we have implemented them in a FPGA (a *Spartan 2E XC2S200E*). The size of the macrocell which implements the transmitter is less than 1,000 equivalent gates, while the receiver uses about 10,000 equivalent gates. Experimental results have been obtained with a spectrum analyzer, fully satisfying the frequencial specifications. The frequency error is approximately the same as the one estimated with the simulations. Both macrocells have been tested with satisfactory results.

VI. Conclusion

We have designed two completely digital macrocells which implement a transmitter and a receiver of DTMF signaling. The design complies with the international recommendations for these devices. It is also remarkable the small size

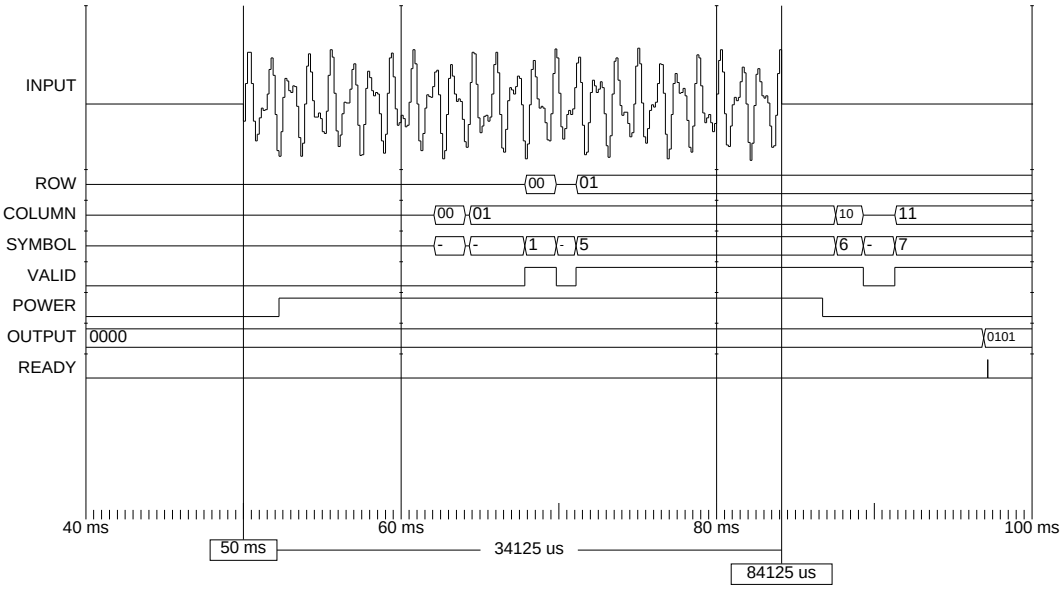

Fig. 7. Minimum symbol duration

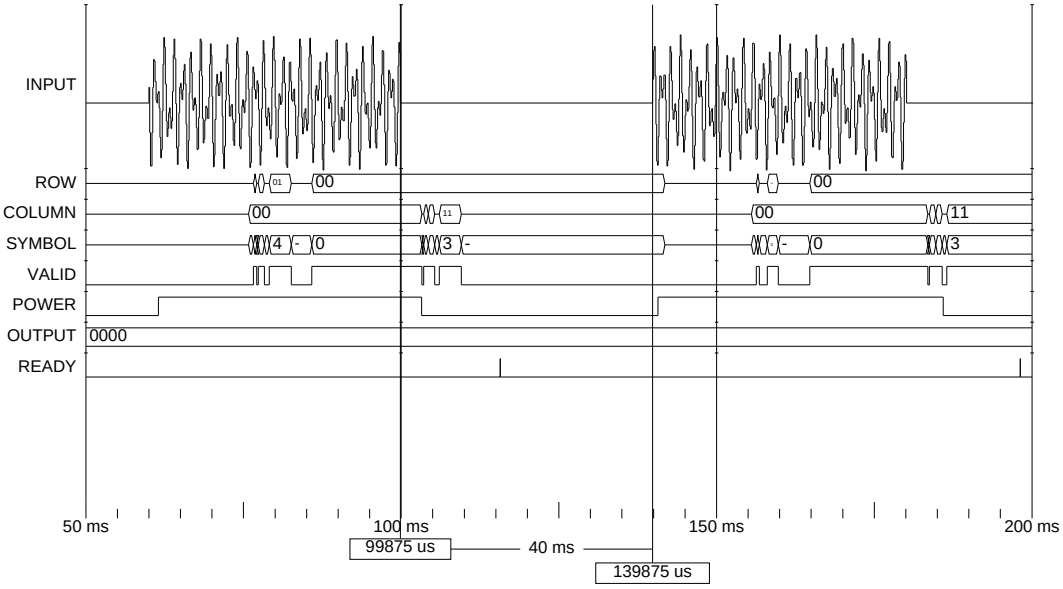

Fig. 8. Pause duration

of the complete design, what makes it suitable for bigger designs which may need a DTMF interface. The receiver is implemented using as mathematical operators just 25 adders/subtractors.

REFERENCES

[1] "A Technical Tutorial on Digital Signal Synthesis," Analog Devices, 1999.

[2] A. Mar, Ed., *Digital Signal Processing Applications Using the ADSP-2100 Family*. Prentice-Hall, 1990, ch. 14, pp. 441–457.

[3] G. Schmer, "DTMF Tone Generation and Detection: An Implementation Using the TMS320C54x," Texas Instruments, Tech. Rep., 2000.

[4] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

[5] M. D. Felder, J. C. Mason, and B. L. Evans, "Efficient Dual-Tone Multi-frequency Detection Using the Nonuniform Discrete Fourier Transform," *IEEE Signal Processing Letters*, vol. 5, no. 7, pp. 160–163, July 1998.

[6] G. Arslan, B. L. Evans, F. A. Sakaraya, and J. L. Pino, "Performance Evaluation and Real-Time Implementation of Subspace, Adaptive, and DFT Algorithms for Multi-tone Detection," in *Proceedings IEEE International Conference on Telecommunications*, April 1996, pp. 884–887.

[7] A. A. Deosthali, S. R. McCaslin, and B. L. Evans, "A Low-Complexity ITU-Compliant Dual Tone Multiple Frequency Detector," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 911–917, March 2000.

[8] "Recommendation Q.24: Multi-Frequency Push-Button Signal Reception," ITU Blue Book, 1989.

[9] V. Friedman, "A Zero Crossing Algorithm for the Estimation of the Frequency of a Single Sinusoid in White Noise," *IEEE Transactions on Signal Processing*, vol. 42, pp. 1565–1569, June 1994.