J.King Saud Univ., Vol.7, *Comp.& Inf.Sci.*, pp.49 - 67 (A.H 1415/1995).

A Real-Time Arabic Text-to-Speech System Based on Parallel Processing

H.A. Fathi*, M.F.Abou El-Yazzed, S.A.Mashali*, and M.R.El-Ghonemy****

* *Electronics Research Institute, and*

** *Electronics and Communications Dept., Faculty of Engineering, Cairo University, Giza, Egypt*

(Received September 1993; accepted for publication 18 April 1994)

Abstract. In this paper, a real-time Arabic text-to-speech system is presented. There are many Arabic text-to-speech systems already built. K.H. Abdel-Gawad and M.F. abou El-Yazeed reported some of them. However, they are not real-time ones. Real-time performances is achieved by using parallel processing techniques. The proposed system consists of a personal computer (PC) and an interface card, to be inserted inside the PC, which carries an N80196KC processor with all logic circuits required to operate the two processors together. The choice of a PC to implement the system is due to its wide spread all over the world which makes this system available at everyone hand. The details of how the two processors cooperate to finish their assigned jobs is given. Although the proposed system is designed to satisfy text-to-speech requirements, with some modification it can be made a *general-purpose* one with modular structure.

1. Introduction

Digital signal processing applications are frequently computationally intensive and require too much time to implement a certain algorithm. Although special digital signal processors were designed with high accuracy and fast computation rate [3-6], they are not enough, as stand alone systems, to implement a real-time system. In order to reach real-time performance, many of the computations must be carried out concurrently. Many systems were designed and built which utilize parallel processing techniques indigital signal processing applications [7-10]. Some of these systems [10] use special-purpose VLSI chips in their design. Although special-purpose chips are faster than general-purpose ones, they are limited to the application for which they were designed. Also, they are more expensive because of the limited size of production. The system in [9] is

implemented on a supercomputer, which is very expensive and impractical for public use.

Section 2 explains the text-to-speech conversion process, section 3 describes the basic system configuration while section 4 presents the details of the implementation. Experimental results are given in section 5 and finally the conclusion comes in section 6.

2. Text-to-Speech Conversion

A text-to-speech system accepts a written text as an input and generates speech as an output. The importance of such a system appears in many applications like reading machines for blind people, receiving electronic mail by phone, accessing large databases by telephone, and many others.

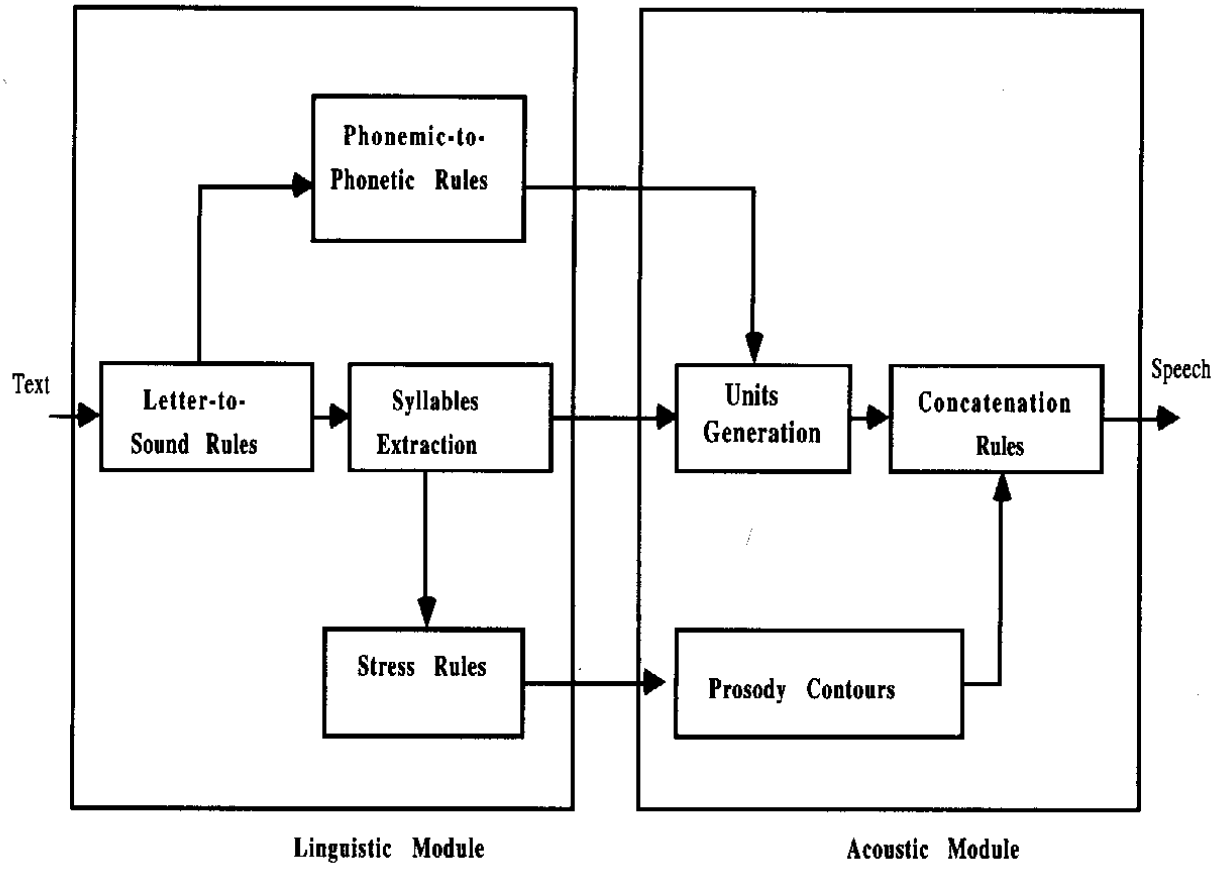

Fig.1. Block diagram of text-to-speech conversion process.

In general, a text-to-speech system consists of two major parts: the linguistic processor and the speech synthesis processor. The linguistic processor converts the input text into phonemes, stress marks, and synthetic stress indicators. In the synthesis process, the prosodic features, phoneme duration, pitch, and intensity, are first generated. Then, the synthesis units are extracted from a phonetic inventory and the concatenation rules applied to produce the final pitch and energy contours. A block diagram of the system is shown in Fig.1.

The synthesis units can be stored as time samples or in some coded form. The first method has the disadvantage of huge storage space requirement but uses simple concatenation rules. The second uses complex rules to produce continuous speech but requires small storage space. We used the linear predictive coding (LPC) of speech to store the synthesis units.

2.1 The linguistic processor

Each language has its own linguistic features so, it needs its own linguistic processor. The complexity of the linguistic model depends on the relation between the oral form of the language and its written form. In some languages, like Arabic and Spanish, there is a close relation between spelling and pronounciation and hence the implementation of the processor is easier than in English and French. The main functions of the processor are[11]:

1) Grapheme-to-phoneme transcription,
2) segmentation of the input text into words,
3) converting words into syllables,
4) generation of prosodic symbols necessary to synthesixe the prosodic features of speech, and
5) syntactic, semantic, and discourse analysis of the input text.

2.1.1 Graphemic-to-phonemic transcription

The firststep in text-to-speech conversion is the transcription of the input text into phonemes. A special Arabic editor was built to accept the input text [2]. This input text is a sequence of characters which may be letters, marks, or delimiters. The marks are short vowels, such as *Fatha, Kasra, and Damma, Sukun and* diacritics. Diacritics are used to represent some morphophonemic processes of the language like the gemmination mark *Shadda, Tanween, and El-madd* marks. Some delimiters are used to indicate the end

of a sentence "." or to distinguish the linking between two adjacent words "-". Mapping the orthographic input into phonemes is done using Table 1.

Table 1 Arabic phonemes and its phonetic symbols.

Arabic phoneme	Phonetic symbol
ء	/P/
ب	/B/
ت	/T/
ث	/V/
ج	/G/
ح	/H!/
خ	/X/
د	/D/
ذ	/C/
ر	/R/
ز	/Z/
س	/S/
ش	/S!/
ص	/S&/
ض	/D&/
ط	/T&/
ظ	/C&/
ع	/A!/
غ	/G!/
ف	/F/
ق	/Q/
ك	/K/
ل	/L/
م	/M/
ن	/N/
ه	/H/
و	/W/
ى	/Y/

الف مد	آ	/A%/	
ياء مد	ي	/E%/	Long vowel
واو مد	و	/O%/	
فتحه	ـَ	/A/	
كسرة	ـِ	/E/	Short vowel
ضمة	ـُ	/O/	

2.1.2 Segmentation of the input text into words

Arabic text is composed of characters with spacing between words. By tracking these spaces, a simple algorithm for word segmentation can be implemented. In Arabic language, forming the phonetic transcription of a sentence is much more complicated than just concatenating the transcription of its word components [2]. The linking problem appears at the boundary between two suscessive words if the second begins with the definite article *Alef lam* or the letter *Alef* representing *Hamzet El-Wasl.* Deciding whether the *Alef lam* is a definite article or not is difficult to the computer as semantic analysis must be done if automatic distinction is to be done. So the editor uses a special delimiter "-" to overcome this problem.

2.1.3 Converting words into syllables

A syllable is a speech segment that includes a nuclear vowel and neighboring consonants. The advantage of such long speech unit is that it includes the coarticulation effects between neighbor segments. That is why we used syllables as our synthesis unit. In Arabic language, there are five types of syllables [12].

1) Consonant + short vowel, denoted by CV.
2) Consonant + long vowel, denoted by CV%.
3) Consonant + short vowel, consonant, denoted by CVC.
4) Consonant + long vowel, consonant, denoted by CV%C.
5) Consonant + long vowel, 2 consonants, denoted by CV%CC.

The major disadvantage of syllables is the large storage space required. The vowel in all these patterns, which is the nucleus of the syllable, is always the second phoneme. This property allows word syllabification to be done easily. Word syllabification is required because it allows the specification of some suprasegmental aspects of speech. In Arabic, the lexical stress, which is one of the major determinants of speech prosody, is based entirely on word morphology (number of syllables, its type, distribution, ...etc.).

2.1.4 Generation of prosodic marks

In the synthesis process, the pitch, intensity, and length (in time) of each syllable are affected by the stress (energy) and type of the sentence. There are two types of stress; namely *lexical* or *word* stress and *sentential* or *phrasal* stress. Lexical stress is applied on one of the word syllables while sentential stress is applied on one or more words in a sentence. The rules of the lexical stress are language dependent [2].

Stress can also alter the meaning of a word; i.e., two words having the same graphemic to phonemic transcription. The sentential stress is denoted by a semantic marker. A question may have different semantics according to the stressed word in the question. Another marker is necessary to indicate the sentence type; i.e., declarative, imperative, or question.

2.1.5 Speech prosody

Prosodic features are those which characterize an utterance as a whole, rather than having a local influence on individual sound segments. In computer speech, an utterance consists of a single unit of information which is the sentence. In natural speech, an utterance can be very much longer, but it is broken into prosodic units which are roughly sentences. These prosodic units are closely related to each other.

Variations in voice dynamics occur in three dimensions: pitch, time, and amplitude [13]. These dimensions are inextricably twined together in natural speech. Variations in voice quality are much less important for the factual kind of speech usually sought in voice response applications, although they can play a considerable role in conveying emotions of the speaker.

Early speech synthesis systems did not pay too much attention to prosodics. This was so for the main goal was to provide intelligible speech which can be understood clearly. In order to produce natural speech from a text-to-speech system, it is not adequate to concatenate a sequence of phonetic units from a stored dictionary. Prosody enables the listener to segment the synthesized speech into words and phrases and thus to understand it [14]. The main prosodic features of speech are intonation, stress, mood, contextual nasalization, and other suprasgmental events.

Each language has its own prosodic structure which needs extensive study of the language to extract its main features. Since no such work was done for the Arabic language, previous work on speech synthesis did not take prosodic features into consideration. The system implemented in [2] related prosodic features to syllabic structure. The reasons for this are:

a) The lexical stress rules are found to be based entirely on word morphology (its syllabic structure).

b) Investigation of prosodic features for different words having the same syllabic structure shows a great similarity on prosodic contours (assuming that they are pronounced with the same mood). Only context dependence should be taken into consideration.

The synthesis of the prosody contour was made in two main steps. In the first, word level, preliminary contours for words based on their syllabic structure are generated. In the second, a global semantic contour is superimposed on these word contours.

3. System Architecture

The design of the system presented here is of the MIMD (Multiple Instruction Multiple Data) closely-coupled type [15]. In this type, the two processors operate nearly independent of each other except for the times they need to exchange data between each other. This configuration was selected mainly because of its ease of implementation, using discrete components, as an add-on card for use with the personal computers, and also because of the sixe of the application to be used with. Also the design complexity is greatly reduced especially for the two-processor systems. Another important factor, in the application considered here, is that it is better to partition the job -text-to-speech- between two processors operating relatively independent of each other so that one can make use of the facilities available on the PC; e.g. keyboard interface, monitor, secondary storage devices, etc. In this way, the user can interact directly with the PC while the other processor is producing the speech. Finally, there remains the restriction of the PC design which does not allow external devices to access its main memory.

The proposed system depends on a personal computer (PC) and a general-purpose microcontroller chip **N80196KC**. They work together in a master slave mode of operation, the host computer is the master and the other processor is the slave. There are 4K Byte SRAM (Static Random Access Memory) which can be accessed by both the host and the microcontroller. The data flow in the system is unidirectional; i.e. data transfer is from the host to the microcontroller. This was done to facilitate the control logic of the shared memory between the two processors since the application allows this possibility. The N80196KC and the SRAM together with the required logic are on an IBM AT interface card inserted inside the host PC. A block diagram of the system is shown in Fig. 2. The details of the circuit design will be introduced in the next section.

Microcontrollers are, as the name implies, processors whose internal architecture is optimized for control applications. A microcontroller consists of a processor, on chip RAM, and sometimes, on chip EPROM (Erasable Programmable Read Only Memory) [16]. Also, there are many peripherals contained in the chip, like serial port, A/D (Analog to Digital) converter, timers, counters, and many parallel I/O (Input/Output)

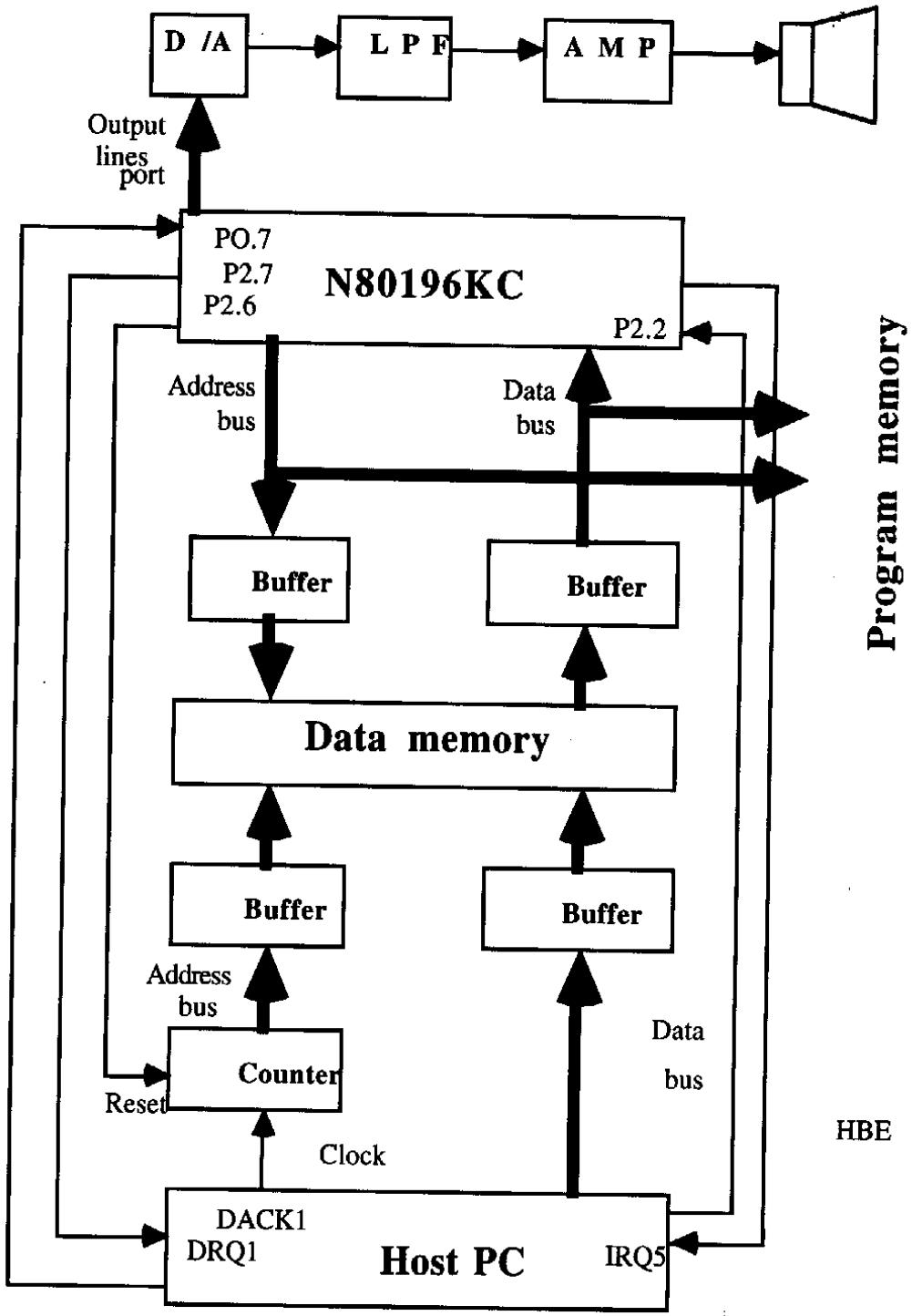

Fig. 2. Block diagram of the proposed system

ports. They operate at high frequencies; e.g. 8,12,16 MHz. Their instruction sets have both byte and bit manipulation instructions. Most of the instructions need one micro second execution time giving throughput of about 1MOPS (million operation per second). The 80C196KC microcontroller has some special features. Its ALU (Arithmetic Logic Unit) is 16-bit wide. Also, it has a register-to-register architecture thus allowing operations to be carried out quickly on any of the 232 registers available on the chip. This avoids the problem of having to move data between the memory and the processor registers to enable operations to be performed on new data. By this way, the processing time can be reduced significantly. The main reasons behind using this processor are: high computation rate, for mid-range digital signal processing applications, register-to-register architecture, on chip output port for sending out the time samples, and cheap off-shelf availabilit.

Since the LPC model used for speech sinthesis is 10th order, we need 10 multiplications and 28 additions per one time sample. This gives a total of 0.304 MOPS for a sampling frequency of 8 KHz. The parameters of the LPC model are in fixed point representation. All the computations of each sample must finish before those of the next one. This is the main reason behind choosing the 80C196KC processor. Also, this is one of the reasons for using sampling frequency of only 8KHz as the microcontroller has only125 use to carry out the required computaions to synthesize one sample would not be sufficient to carry out the required operations. Also, it is sufficient to produce telephone-like quality. This feature allows accessing data bases through the telephone line.

In the proposed system, the main job of the system (text-to-speech) is partitioned between the two processors. The host accepts the input text from the keyboard and carries out all the necessary computations for text analysis; i.e. graphemic to phonemic transcription, segmentation of the input text into words, word syllabification, speech prosody; e.g. intensity, intonation, etc. The output of this phase is the reflection coefficients [17], pitch, and the residual. These are used by the microcontroller to calculate the corresponding time samples. The synthesized frames are pitch synchronous.

The cooperation between the two processors is based on the demand driventechnique [18]. The host computer starts the text-to-speech conversion process where the grapheme to phoneme transcription process is carried out. When the speech output is to be produced, which is done by the other processor, the host sends an interrupt signal asking for the computation of the time samples. The second processor then starts executing its task. Since the two processors are not of the same type, there must be some way to synchronize their operation. An interrupt-based handshaking technique is used. When the host finishes preparing data, it sends an interrupt signal to the microcontroller indicate that data is ready for synthesizing speech.

4. Hardware Description of the System

As was mentioned in the previous section, the interface card carries the microcontroller, the common memory, the required logic for controlling data transfer between the two processors, the D/A (Digital to Analog) converter, the low-pas filter, and the power amplifier. Also, there is an 8K Byte of EPROM which contains the program to be executed by the microcontroller. The EPROM is accessed by the microcontroller only so it is connected directly to its data and address buses while the shared memory (SRAM) is connected to both the host and microcontroller through buffer stages. The schematic diagram of the implemented system is shown in Fig. 3.

At the time when the common memory is connected to the host it is disconnected from the microcontroller and vice versa. The logic controlling this process is generated by the microcontroller through an output port line P 2.7. When it is low, the host can write data into the memory and when high the data can be read by the microcontroller. This signal is kept low untill the last byte in the array of data is moved from the host memory to the shared memory. The buffer between the common memory address bus and the microcontroller address bus is controlled by an address line (AD13) of the microcontroller. On the other hand, the buffer between the two data buses is controlled by a delayed version of the RD signal, EN, from the microcontroller. This delay (about 40ns) was necessary to adjust the bus timing. A wait state is inserted in the system bus cycle of the microcontroller to allow enough time for read operation from the SRAM to be completed.

Since the host communicates with the shared memory as an output port (using DMA technique), an external counter is used to generate the address of the memory. This counter is clocked by the DACK1 (Data Acknowledge of channel 1) signal generated from the host computer. However, as this signal is active low and the counter is negative edge triggered, an inverter is used to complement the counter clock so that it is incremented only after the byte transfer operation takes place. After finishing data transfer, the counter is reset to zero by the controller output port line P2.6.

When data is ready in the common memory, the data request signal is set inactive and the address and data buses of the host are isolated from the common memory to allow the controller to read the necessary data. When half of the data in the shared memory is used up, the microcontroller sends an interrupt request signal to the host (IRQ5), through a latch, to hold the interrupt for enough time to be detected by the host, asking for new data. The interrupt service routing on the host responds by preparing data for the transfer operation, clearing the latch to allow further interrupts to be received, and sending an interrupt to the microcontroller. The service routine for that interrupt on the microcontroller raises the DMA request signal and the buffers between the host bus and

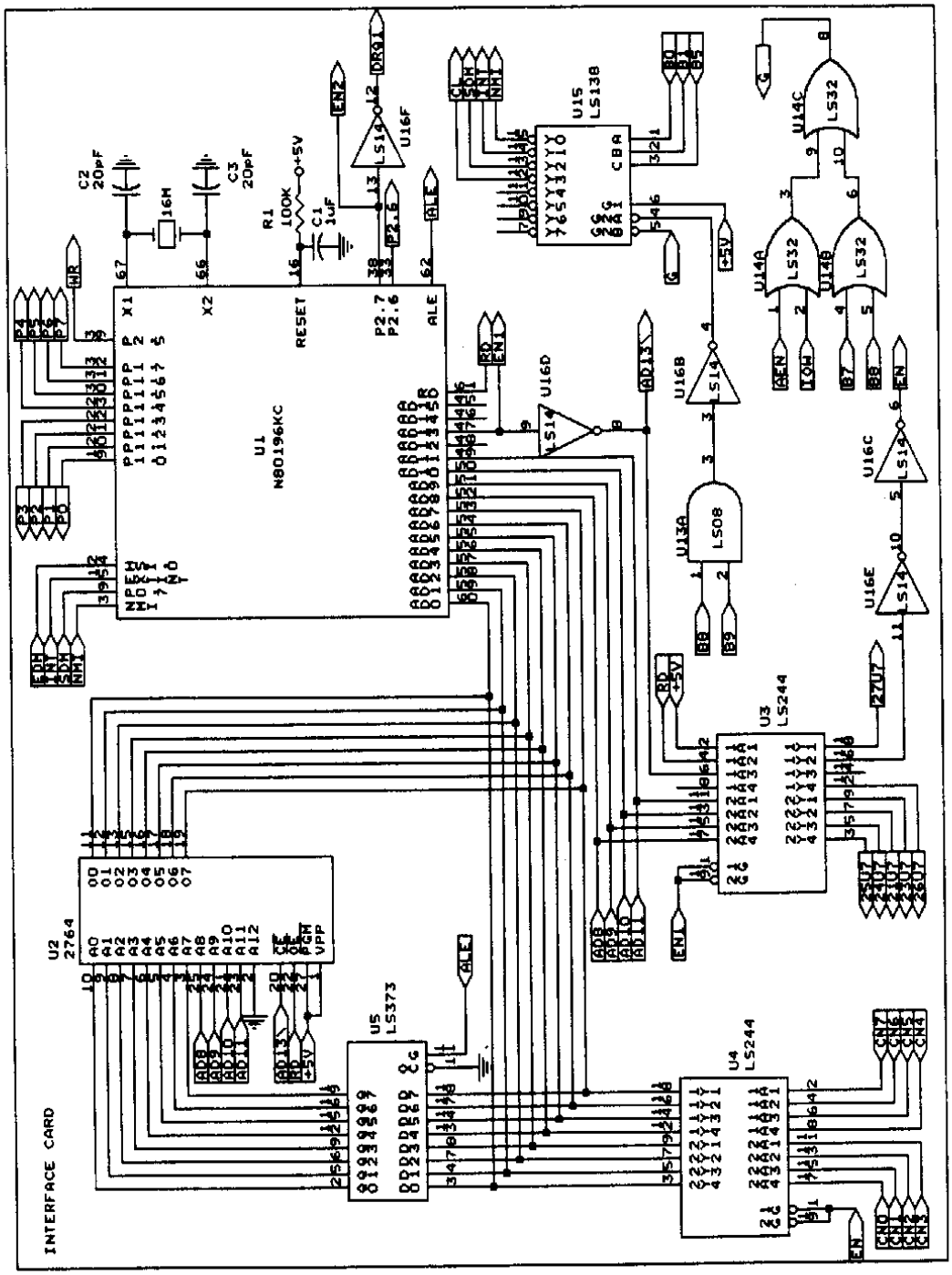

Fig. 3a The schematic diagram of the implemented hardware system

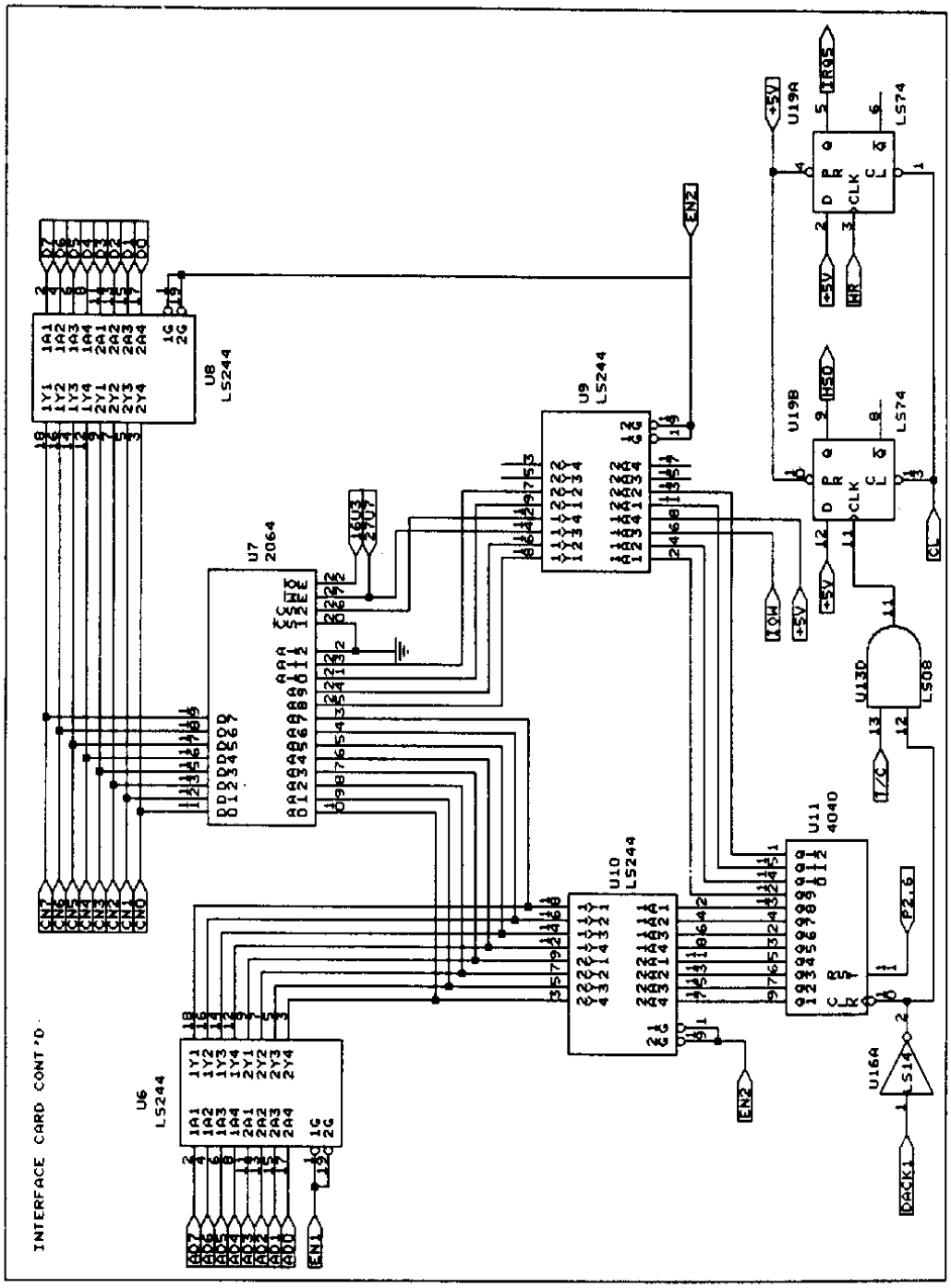

Fig. 3b The schematic diagram of the implemented hardware system

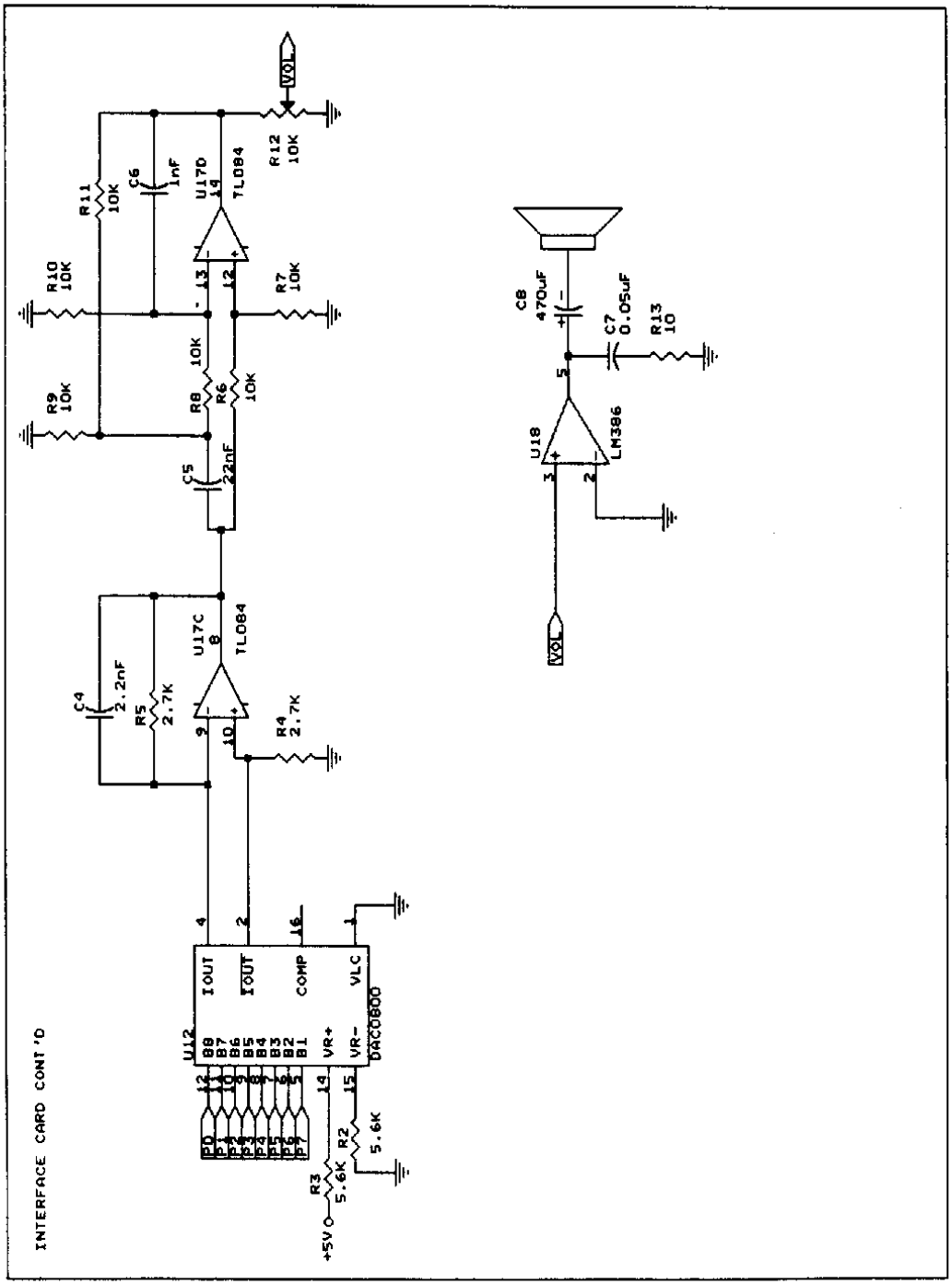

Fig. 3c The schematic diagram of the implemented hardware system

the common memory are enabled. This routine also checks which half of the memory is to be filled with data. If it is the top half, the external counter is set to zero, otherwise, it is left unchanged to allow data to be written in the bottom half. In this way, the shared memory looks to the microcontroller as a circular buffer which allows speech of unrestricted length to be produced continuously. During the transfer operation, the microcontroller is busy calculating one frame of speech and is in no need to access the common memory. At the end of the DMA cycle, an interrupt, which is generated through a latch that is driven by the T/C (Terminal Count) signal and the inverted DACK1 signal ANDed together, is sent to the microcontroller to stop the DMA request. In order to prevent any pauses in the speech synthesis operation, which is not acceptable, both interrupts on the microcontroller (starting DMA operation) are enabled only when data necessary for synthesizing one frame of speech is already available to the microcontroller in its internal RAM. In other words, priority for the access of the shared memory is given to the microcontroller, READ operation, over the host PC, WRITE operation. This also overcomes the multiple access to the shared memory problem.

Since the time samples calculated by the microcontroller are in digital form, a DAC (digital to analog converter) is used to produce the analog speech signal. An 8-bit DAC (DAC0800) will give enough accuracy for the resulting speech signal. Following the DAC stage there is a third order LPF (Low Pass Filter) whose cutoff frequency is 3.4 kHz to prevent aliasing effect and provide clear sound. The last stage is a power amplifier which drives an 8 ohm loud speaker.

5. Results

Now, we introduce the results of both the quality of the synthesized speech and speed of our system. Table 2 lists these results. In the intelligibility test, sentences are heard by ten subjects. Subjects are not informed of the sentences to be heard. Results show that, in most cases a success of recognition from the first trial is achieved. In cases where there is a failure to recognize any of the sentence words from the first trial, repetitve trials, limited to five, are made. The text of the synthesized speech is then provided and subjects are asked to indicate all synthetic units that are not intelligible to them. One point of interest is that, the score for sentences is higher than that for isolated words. That is expected as subjects use their own experience in language to predict the contents of synthetic speech.

Table 2.A. Results of words intelligibility test

Subject	1	2	3	4	5	6	Av.
Success Rate	68.3%	73.3%	65.0%	71.7%	91.7%	76.6%	74.4%

Table 2.B Results of the sentence intelligibility test

Trial No.	Recognition Rate										Av.
	1	2	3	4	5	6	7	8	9	10	
1	90%	100%	90%	80%	50%	80%	60%	60%	80%	90%	88%
2	90%	100%	90%	90%	60%	100%	80%	100%	90%	100%	90%
3	90%	100%	90%	90%	70%	100%	90%	100%	90%	100%	93%
4	90%	100%	100%	90%	80%	100%	100%	100%	90%	100%	93%
5	90%	100%	100%	90%	80%	100%	100%	100%	90%	100%	95%

As stated before, the text-to-speech processconsists of two parts; text analysis and speech synthesizing . After the text has been analysed, the computer starts the synthesis process. During that time, the computer is tied in calculating the time samples and sending them to an output port and no other computations can be carried out. Thus the computer stays almost idle until the written message has been pronounced. This time depends on the length of the sentence; i.e. number of words and word length. This was the case in the system implemented in [2].

However, this is not the case in the system presented here. The job is partitioned between the two processors, the host PC and the microcontroller. When the host finishes the analysis of the input text, it sends the corresponding speech data to the second processor. Then, the second processor starts the speech synthesis process and the host is now free to accept new text. This is what we mean by real-time. Figure 4 shows the timing diagram for the single-processor and two-processor cases applied to three sentences. In this figure, T_{ai} is the time required by the host PC to analyze the *ith* sentence and T_{si} is the time require by the microcontroller to synthesize the corresponding time samples. In the single-processor case, the total time is given by:

$$T_{tot1} = T_{a1}+T_{s1}+T_{a2}+T_{s2}+T_{a3}+T_{s3} \tag{5.1}$$

or in general

$$T_{tot1}= \sum(T_{ai}+T_{si}) \qquad ,i=1,2,..., N. \tag{5.2}$$

where N is the number of messages (sentences). In the two-processor case, the total time is given by:

$$T_{tot2} = T_{a1}+T_{a2}+T_{a3}+T_{s3} \tag{5.3}$$

or in general

$$T_{tot2} = \sum(T_{ai}) + T_{SN} \qquad ,i=1,2,..., N. \tag{5.4}$$

Thus the total reduction in time is given by:

$$T_{red} = \sum T_{si} \qquad ,i=1,2,...,N-1. \tag{5.5}$$

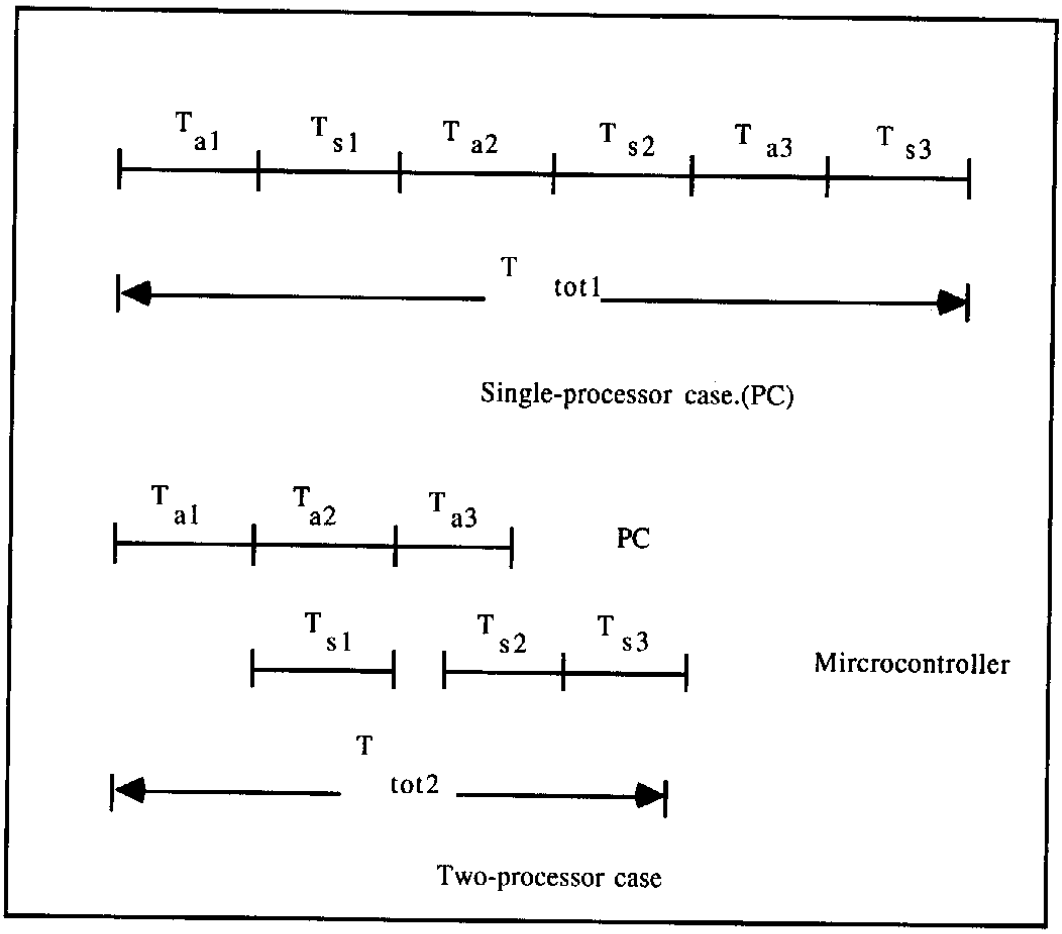

Fig. 4. Timing diagram of the two systems.

The gap between T_{s1} and T_{s2} means that the analysis time of sentence 2 is longer than the synthesis time of sentence 1. Some examples have been worked out and the results showed that the reduction in time is between 20% and 30% on a PC/AT machine running at 16MHz clock while this ratio is between 35% and 60% on a PC/486 system running at 33MHz clock.

6. Conclusion

A real-time Arabic text-to-speech system has been introduced. The proposed system consists of two processors, the host PC and an N80196KC microcontroller, operating together. The Arabic editor, which accepts the input text, and all processing necessary to get the reflection coefficients corresponding to the input text are developed on the PC by a high-level language, PASCAL. The synthesis program, to be executed by the microcontroller is developed on the PC. An assembler is used to convert the assembly language program into the machine code.Then another program relocates this code into the right locations on the EPROM. The presented system allows the user to listen to a sentence while he, or she, is entering a new one. The resulting speech is of good intelligibility and naturality levels. Although the design was directed to the text-to-speech application, it is a general-purpose one and only little modifications are required. Also the design is modular allowing more processors to be added to provide higher computational through out. Despite that there exists many speech synthesis chips available, we did not use any of them as the intention was to construct a general-purpose system cabable of increasing the computation power of the PC. Also these chips need frequent communication with the PC due to its small queue thus producing overhead on the PC execution time.

References

[1] Abdel-Gawad, K.H. "Arabic Text-to-Speech System." *M.Sc. Thesis*, Cairo Univ.1989.

[2] Abu-Elyazeed, M.F. "An Arabic Text-to-Speech System." *Ph.D. Thesis*, Cairo Univ., 1990.

[3] Morris, L. R. " Digital Signal Processing Microprocessors: Forward to the Past? " *IEEE Micro*, 6, (Dec. 1986), 6-8.

[4] Shivley, R. R. "Architecture of a Programmable Digital Signal Processor." *IEEE Trans. Comput.*, C 31 (Jan. 1982), 16-22.

[5] Glass, M. " A Programmable Signal Processor Architecture."*Proc. ICASSP*, (1979), 702-05.

[6] Boddie, J. R., *et al.* "Degital Signal Processor : Architecture and Perform-ance." *Bell Syst. Tech. J.*, 60, Part 2 (Sept. 1981), 1449-62.

[7] Pawate, B. I.,*et al* . "Connected Word Recognizer on a Multiprocessor Sys-tem." *Proc. ICASSP*, (1987), 1151-54.

[8] Tang, G.Y. and Lien, B.K."A Multiple Mircroprocessor System for General DSP Operation." *Proc. ICASSP*, (1987), 1047-50.

[9] Kimball, O., *et al.* " Efficient Implementation of Continuous Speech Recognition on a Large Scale Parallel Processor." *Proc. ICASSP*, (1987), 852-55.

[10] Murveit, H. and Brodersen, R.W."An Integrated-Circuit-Based Speech Recognition System." *IEEE Trans. ASSP*, 34, No. 6 (Dec. 1986), 1465-72.

[11] Allen, J. "Synthesis of Speech from Unrestricted Text." *Proc. IEEE*, 64 (1976) , 433-42.

[12] Anees, I. " *Arabic Sounds*." Cairo : Dar El-Nahdah El-Arabeyah, 1961.

[13] Haggard, M.P.; Mbler, A. and Callow, M. "Pitch as Voicing Cue." *JASA*, 47, (1970), 613-17.

[14] Allam, A. A. "From Timing in Arabic Languages." *Ph.D. Thesis*, El-Azhar Univ., Cairo, 1970.

[15] Hwang, K. and Briggs, F.A.*"Computer Architecture and Parallel Process-ing."* New York: McGraw-Hill, 1984.

[16] "16-bit Embedded Controller." *Intel Corp.*, 1991.

[17] Makhoul, J. "Linear Predictive Coding." In *: Electronic Speech Synthesis: Techniques, Technology, and Applications*. London. Granada Publishing Ltd., 1984.

[18] Gajski, D.D. and Peir, J. "Essential Issues in Multiprocessor Systems." *Computer*, 18, No. 6 (June 1985), 9-25.

نظام لتحويل النصوص العربية إلى كلام منطوق في الزمن الحقيقي مؤسس على المعالجة المتوازية

حسن فتحي *، محمد فتحي أبواليزيد* *، ساميه عبدالرزاق مشالي *
محمد رياض الغنيمي* *

* *قسم الحاسبات والنظم، معهد بحوث الالكترونيات،*
شارع التحرير- الدقي - الجيزة، جمهورية مصر العربية
* * *قسم هندسة الالكترونيات - كلية الهندسة، جامعة القاهرة*
الجيزة ، جمهورية مصر العربية

ملخص البحث . يقدم البحث نظاماً لتحويل النصوص العربية إلى كلام منطوق في الزمن الحقيقي . وهناك عدة نظم تم بناؤها فعلاً لتقوم بهذه المهمة . وقد بين عبدالجواد و أبو اليزيد بعض هذه النظم ولكن هذه النظم، على أية حال، ليست نظم زمن حقيقي . ويمكن تحقيق الأداء في الزمن الحقيقي باستخدام أساليب المعالجة المتوازية .

ويتكون النظام المقترح من حاسوب شخصي ذي بطاقة مواجهة بينية موضوعة بداخله، وتحمل هذه البطاقة معالجاً من طراز N80196KC مع كافة الدراسات المنطقية اللازمة لتشغيل المعالجين معاً . وقد تم اختيار الحاسوب الشخصي لتنفيذ هذا النظام بناء على انتشاره الواسع في العالم مما يجعل هذا النظام ميسوراً لمن يحتاجه . ويبين البحث كيفية تعاون المعالجين لتنفيذ ما يسند إليهما من مهام .

و رغم أن النظام المقترح صمم ليقابل احتياجات تحويل النصوص إلى كلام منطوق إلا أنه، وببعض التعديلات، يمكن أن يكون نظاماً صالحاً لأغراض عامة ذا بنية نمطية .