

# Building speech recognition into portable products

Stanley Kubrick's movie, "2001: A Space Odyssey," introduced us to HAL, a computer that could understand pretty much every language in the universe. Now that we are getting close to the time frame featured in Kubrick's movie, we must admit that we are still quite a bit separated from the capabilities of HAL. However, we are getting there very fast.

There is a large variety in the speech-recognition technology and it is important for you to understand the differences. You can classify speech-recognition systems according to the type of speech, the size of the vocabulary, the basic units and the speaker dependence. The position of a speech-recognition system in these

dimensions determines which algorithm can or has to be used.

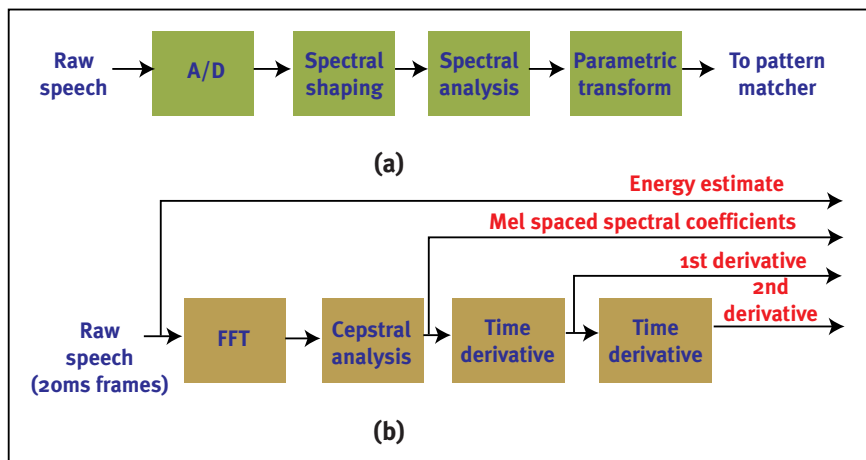
One of the most important characteristics in automatic speech recognition (ASR) lies in the type of the speech. There are basically two types of speech: continuous speech and discrete speech. Discrete speech consists of isolated words, which are separated by silences (usually 100ms or more), while with continuous speech, words will be spoken without silences. The advantage of discrete speech is that the word boundaries can be determined exactly.

The size of the vocabulary is the second typical aspect of a speech-recognition application. The vocabulary is the set of words that have to be recognized. A small vocabulary is one that contains less than about 30 words. A 500-word vocabulary is average size. A vocabulary with more than 25,000 words generally will be seen as very big, although these definitions tend to depend on the application field.

Speech recognition systems also differ in the choice of basic speech units. For example, words, syllables or phones—tones from which words are built up—can be chosen as a basic unit. In English, there are about 50 different phones (**table 1**). In practical applications, you would combine two or three phones as a basic unit, called biphones or triphones. This gives you a "co-articulation" effect: the sound

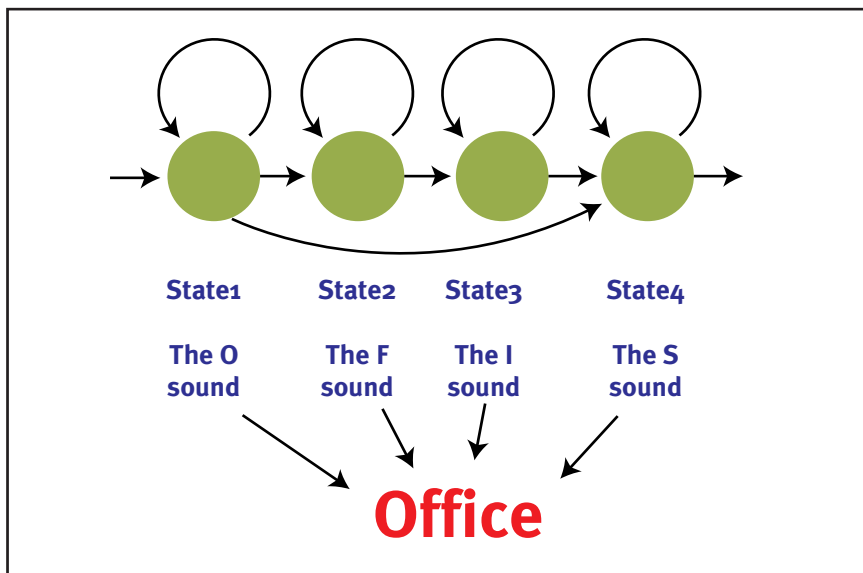
## BY DIRK DEVISCH

Business Development Manager  
Frontier Design Asia Pacific



**Figure 1:** (a) The acoustical front-end conducts data reduction by extracting parameters that accurately represent the incoming speech. (b) An example of a poplar configuration.

SPEECH RECOGNITION



**Figure 2:** In this simplified example, each state represents a letter, and the sequence of letters is modeled as going from left to right throughout the diagram. At each node, there is a certain probability for each possible letter.

of a phone can change depending on the phone that comes before and after it. Biphones and triphones are mostly used in continuous speech recognition.

Speaker independence is the last typical aspect of a speech-recognition application. A system is called speaker independent if the rate of success is independent of the user. A speaker-independent system has to be trained with a number of speakers to represent the spreading in acoustic types of speakers in the vocabulary. Statistical models are used to generate an efficient and sufficiently representative vocabulary.

**Speech-recognition technology**

The architecture of a speech recognition engine can be broken down into a feature extraction module, a pattern matching algorithm, and a hypothesis block. The feature extraction block transforms the input speech into a set of spectral components. The pattern-matching block uses the spectral patterns and compares them with some known patterns. Pattern matching can be based on a wide variety of different techniques, ranging from Dynamic Time Warping (DTW), over

Hidden Markov Models (HMM), to Neural Networks and various combinations of these. Depending on the complexity, the “Pattern Matcher” can also make use of a priori information, such as speech models and language models. Finally, the “hypothesis engine” decides which utterance (word or sentence) was recognized or informs the user of failure to recognize anything.

**Feature extraction**

The feature extraction or acoustical front-end of the ASR system is the

**What’s Online**

► **DIGITAL SPEECH FOR CONSUMER MULTIMEDIA**

Key aspects of speech coding are reviewed with special attention given to coders with low-bit-rate and high-quality speech. Attributes to speech coders, such as delay, complexity, bit rate, and quality, are also discussed.  
[www.ee.asiansources.com/article\\_content.php3?article\\_id=8800009369](http://www.ee.asiansources.com/article_content.php3?article_id=8800009369)

► **OPTIMIZING COMPILERS FOR VLIW MEDIA PROCESSORS**

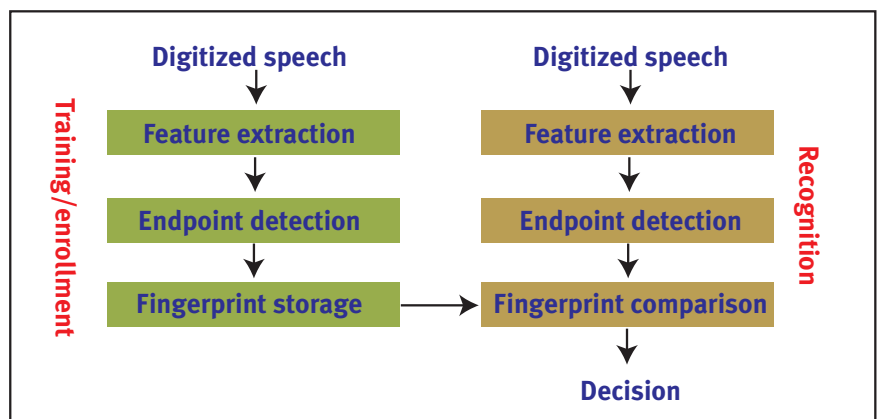
Multimedia processors rely on sophisticated compilers to schedule multiple operations into one long instruction. You can speed up software development and ensure portability using compilers that let you write in C/C++.  
[www.ee.asiansources.com/article\\_content.php3?article\\_id=8800009493](http://www.ee.asiansources.com/article_content.php3?article_id=8800009493)

[www.ee.asiansources.com](http://www.ee.asiansources.com)

first part of the recognizer. The acoustical front-end typically segments the incoming speech signal into 20ms to 30ms frames, which may overlap. Then some data-reduction technique is applied to these frames. This is usually a combination of windowing, LPC analysis and some parametric transform (figure 1).

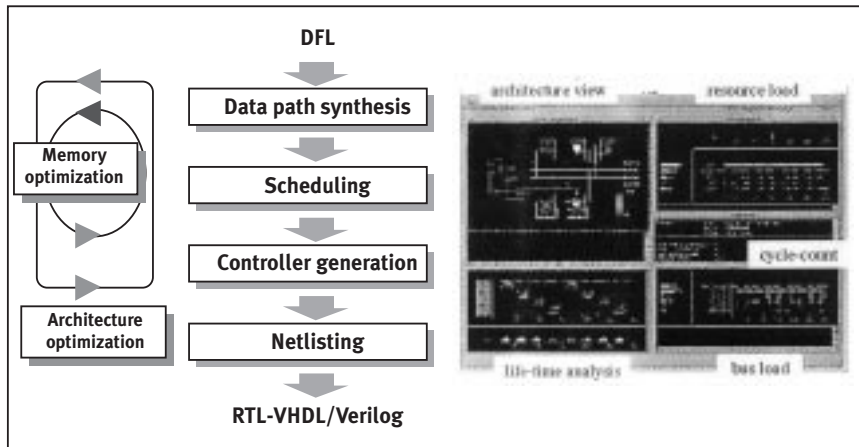
**Pattern matching**

The pattern-matching block tries to match the incoming speech patterns to



**Figure 3:** During the training phase, acoustic templates are computed and stored. During normal operation, the incoming speech is converted into an acoustic template and compared with stored reference templates.

SPEECH RECOGNITION



**Figure 4:** Mistral2 outputs a hierarchical RTL VHDL/Verilog description that can be mapped further into gates by using an ASIC or FPGA design tool.

the existing information in memory. Depending on the type of speech-recognition system, the performance and speaker (in)dependence, a number of different pattern-matching techniques can be used. The pattern-matching algorithm also heavily depends on the speech model used. You can try to match a complete word at a time to existing words in memory, or you can match smaller “units of speech,” such as diphones or triphones.

Systems that match complete words are only used in “isolated-word” speech-recognition systems. Systems that match diphones or triphones are predominantly used in continuous speech recognition. For “word-by-word” pattern matching, the most successful algorithm is DTW. For continuous speech recognition based on biphones or triphones you would typically use algorithms based on HMMs.

DTW is a well-known pattern-matching technique used in isolated-word speech-recognition systems. It basically tries to match the incoming word with the reference word stored in memory. The DTW engine does that by comparing the word piece-by-piece to the reference pattern.

Since not every word is pronounced every time at the same speed and with the same pronunciation, the DTW engine has to sometimes “stretch” the

time axis a little. If you plot the words “SPEECH” sound for sound on the X-axis and the incoming word “SSPEEEhH” on the Y-axis, you will notice that the “path of matching sounds” is not always the ideal straight line. You will also realize that in order to apply DTW to the word-recognition problem, you need to know the start and stop-points of the words. In noisy conditions this is not a trivial task.

The advantages of DTW are:

- Efficient hardware implementations exist.

- The “training” sequence is simple, since it just involves the feature extraction for the words that need to be recognized.

The disadvantages of DTW are:

- It is not suited to continuous speech recognition.
- It requires the computation of the word start and stop points.

**Hidden Markov Models**

HMMs are named after the Russian organic chemist Vladimir Vasilyevich Markovnikov, who introduced them in 1870. They are essentially stochastic processes. There are 2 kinds: Discrete HMM (DHMM) and Continuous HMM (CHMM), the latter mostly used for continuous speech recognition. The principle is illustrated in figure 2.

HMMs represent speech by a sequence of states, each representing a piece of the input signal. The states of the HMM correspond to phones, biphone or triphones. A triphone, for example, is represented by four to five states. For the time being, however, assume that each state corresponds to a letter. At each state there is a probability distribution for

Phones in the English Language					
Phone	Example	Phone	Example	Phone	Example
AA	odd	EY	ate	P	pee
AE	at	F	fee	PD	lip
AH	hut	G	green	R	read
AO	ought	GD	bag	S	sea
AW	cow	HH	he	SH	she
AX	abide	IH	it	T	tea
AXR	user	IX	acid	TD	lit
AY	hide	IY	eat	TH	theta
B	be	JH	gee	TS	bits
BD	dub	K	key	UH	hood
CH	cheese	KD	lick	UW	two
D	dee	L	lee	V	vee
DD	dud	M	me	W	we
DH	thee	N	knee	Y	yield
DX	matter	NG	ping	Z	zee
EH	Ed	OW	oat	ZH	seizure
ER	hurt	OY	toy		

each of the possible letters, and a transition probability to the next state. The speech recognition process then boils down to finding the most probable path through the (pre-computed) network of nodes.

The advantages of an HMM-based approach are:

- It is easy to incorporate other information, such as speech and language models.
- Continuous HMM have been proven powerful for continuous speech recognition.

tion is a major concern. So a highly integrated, single-chip, solution is required. A few implementation alternatives exist:

- An 8bit or 16bit microprocessor, in combination with on-chip ADCs/DACs, on-chip memory, etc.
- Standard DSP and additional memory.
- Custom hardware solution—dedicated logic.

The ultimate in power consumption and price is achieved by using a

Low power, low cost speech recognition core	
Gate count	12k gates
On board memory	40kb RAM (40 words)
Minimum system clock freq.	1.5MHz
Recognition accuracy	98-99%
Package	14 pins, plastic package

Disadvantages of HMM-based systems are:

- It is computationally intensive.
- It requires long training sequences.

Other approaches for pattern matching with large vocabularies and speaker-independent continuous speech, such as neural networks, is still being researched. These new techniques will use a lot of a priori information about the particular vocabulary, language syntax and semantics.

### Isolated-word recognition

Since the technology has become technically feasible and affordable, you see an increasing number of products and applications using speech recognition. Applications include voice dialing with cordless and cellular phones, command and control of home appliances, such as air-conditioners and car stereos, and toys.

In many of these applications, the additional cost and power consump-

tion is a major concern. So a highly integrated, single-chip, solution is required. A few implementation alternatives exist:

### Design methodology

Let us now examine the design methodology using an industrial speech-recognition application—a “voice-controlled money changer”—targeted at the ultra-low cost, low-power market for “give-away-gadgets.”

The first step is to choose a computationally efficient algorithm that gives adequate recognition accuracy, >98 percent (**figure 3**).

Since the goal is to develop an ultra-low power and low-cost, isolated-word, speaker-dependent speech-recognition system, a DTW approach is selected. The system has a training mode and a recognition mode. During the training mode, the system is being trained to recognize words by performing feature extraction, endpoint detection and fingerprint storage.

The endpoint detection is necessary for pattern matching. The endpoint detection is done based on the average energy in the speech signal.

The design flow starts by developing a floating-point version of the algorithm. For this, the DFL language, powerful for DSP algorithm design, has been used. This DFL description has been tested by using time-domain simulation with actual speech input patterns.

The algorithm can be optimized toward a hardware implementation in a dedicated voice-recognition processor. The optimization involves the conversion of the floating-point algorithm into a fixed-point version. Take care in the selection of the number of bits and the quantization and overflow characteristics for each signal in the algorithm. The fixed-point algorithm is verified again with the reference speech patterns.

The next step is the mapping of the algorithm onto a low-power datapath, and the design of the microcode-based controller that drives the datapath. You can automate this design task as well inside the DSP Station environment. For example, the Mistral2 behavioral synthesis tool allows you to create a user-definable datapath and then automatically schedules the speech-recognition algorithm onto that datapath, thereby generating the microcode-based controller (**figure 4**).

The isolated-word, speaker-dependent, speech-recognition algorithm results in the hardware described in **table 2**.

The speech recognition core has been tested against the Texas Instruments' Control & Command Test Set and achieves an accuracy of over 98 percent. ●

You may e-mail your comments on this article to Dirk Devisch at [dirk\\_devisch@frontierd.com](mailto:dirk_devisch@frontierd.com) or fax: 81-3-37175012.