

# ***M•CORE Processor with On-Chip FLEX™ Decoder For Messaging Applications***

*David Ruimy Gonzales  
Senior Member of Technical Staff  
Motorola M•CORE Technology Center  
Austin, Texas*

## ***Introduction***

Pagers have become quite popular in the last 10 years and have led to the introduction of a variety of new portable applications which communicate digital messages to central base stations that dispatch the information to one or more locations. Using a de-facto industry standard for communicating messages, new products are emerging such as automatic utility meter readers, smart vending machines which report inventory or service requests, and automobile security systems which can disable the engine and/or report where the automobile is.

Applications may be remotely situated and require efficient power management. The focus of this paper is to provide a general overview of the Motorola FLEX™ One-Way paging protocol and describe a low power micro-RISC processor based on an M•CORE architecture that implements a one-way FLEX decoder. A general discussion on the FLEX Stack One-Way Software protocol will be covered with emphasis on architectural requirements.

## ***FLEX™ Wireless Transport Protocol***

The entire paging process is based on the production of some form of selective signaling (coding) format that will initiate an alert in one specific receiving device (or possibly a group of these devices). It also provides a format in which to deliver some form of message to that device, the pager. The key element in this process is a format that performs the transfer of information in an efficient and effective manner.

Historically, several formats have evolved to provide this service such as an early analog two-tone and 5/6 tone sequential formats. While effective, these formats had limitations on system capacity, throughput, and functionality. To satisfy the demands of the increasing popularity of paging service, the industry turned to true digital formats. Several formats evolved from this need, such as the Golay Sequential Code (GSC) and the Post Office Code Standardization Advisory Group (POCSAG) protocols. This evolution in signaling formats once again has been driven by the increasing popularity of paging as a personal communications tool.

Motorola's FLEX Protocol is the direct result of research and development by the acknowledged international leader of paging infrastructure design. This protocol is the de-facto global standard for high-speed paging. The reason for this is simple. The FLEX Protocol supports increased transmission speed and capacity. Service providers have adopted the FLEX Protocol because it quadruples the capacity of other paging protocols and significantly improves messaging reliability.

Thus, the FLEX family of wireless transport protocols greatly enhances the channel efficiency and the cost of traditional paging systems while enabling new value-added wireless services. There are two Messaging Systems currently defined, the FLEX™ (one-way data messaging) protocol and the ReFLEX™ (two-way data messaging) protocol.

FLEX based protocols provide higher transmission speeds than older protocols. The FLEX protocol builds upon existing systems using POCSAG 1200 and runs side by side with POCSAG on one RF channel. Current 1200 bits per second (bps) POCSAG systems have a channel capacity of approximately 120,000 numeric pagers per channel. The FLEX protocol provides paging speeds up to 6400 BPS to allow more than 600,000 numeric pagers per channel.

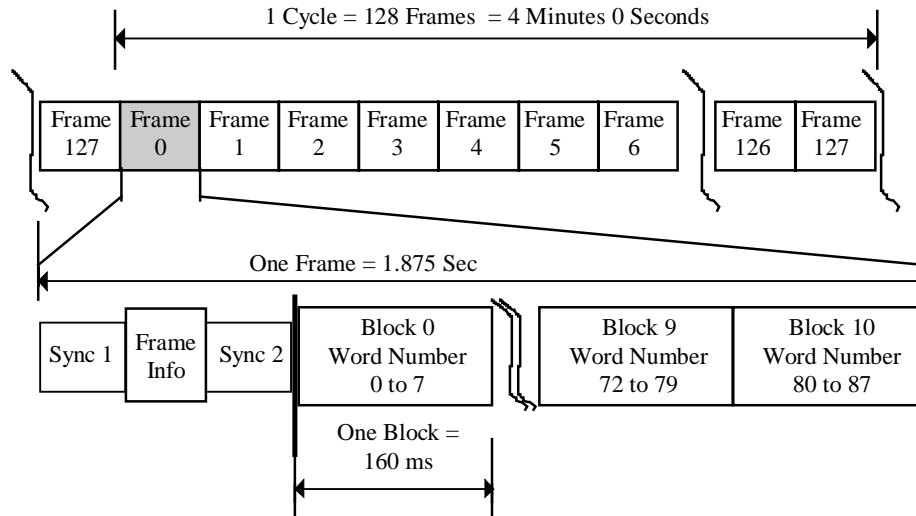
The FLEX protocol maintains data integrity by providing error protection against multi-path fading errors (caused by multi-casting), and by keeping the data-reception electronics continuously in synchronization with the transmission. The FLEX protocol also provides roaming capabilities and significantly improves product battery life.

### ***FLEX Signal Structure Overview***

The FLEX paging protocol is a synchronous time-slot protocol that is referenced to an accurate real-time base, Global Positioning System (GPS). Each pager is assigned to a base frame in a set of 128 frames (0-127) transmitted during a 4 minute time interval called a cycle (32 frames per minute, 1.875 sec. Per frame).

Fifteen FLEX cycles (numbered 0-14, cycle 0/frame 0) occur each hour and are synchronized to the start of the GPS hour. The pager capcode defines its base frame assignment. How often the pager awakes to receive frame information is determined by the collapse value assigned to it that affects battery life.

Each FLEX frame consists of a synchronization and data portion as illustrated in Figure 1. The synchronization portion consists of a synchronization signal and an 11-bit frame information word that allows the pager to identify the frame and cycle in which it resides uniquely. A second synchronization signal indicates the rate at which the data portion is transmitted (i.e., 1600, 3200, or 6400 bits per second). Data is transmitted using either 2-level frequency shift keyed (FSK) modulation or 4-level FSK modulation. The 6400 bps rate is transmitted as four concurrent phases of information using 4-level FSK modulation.



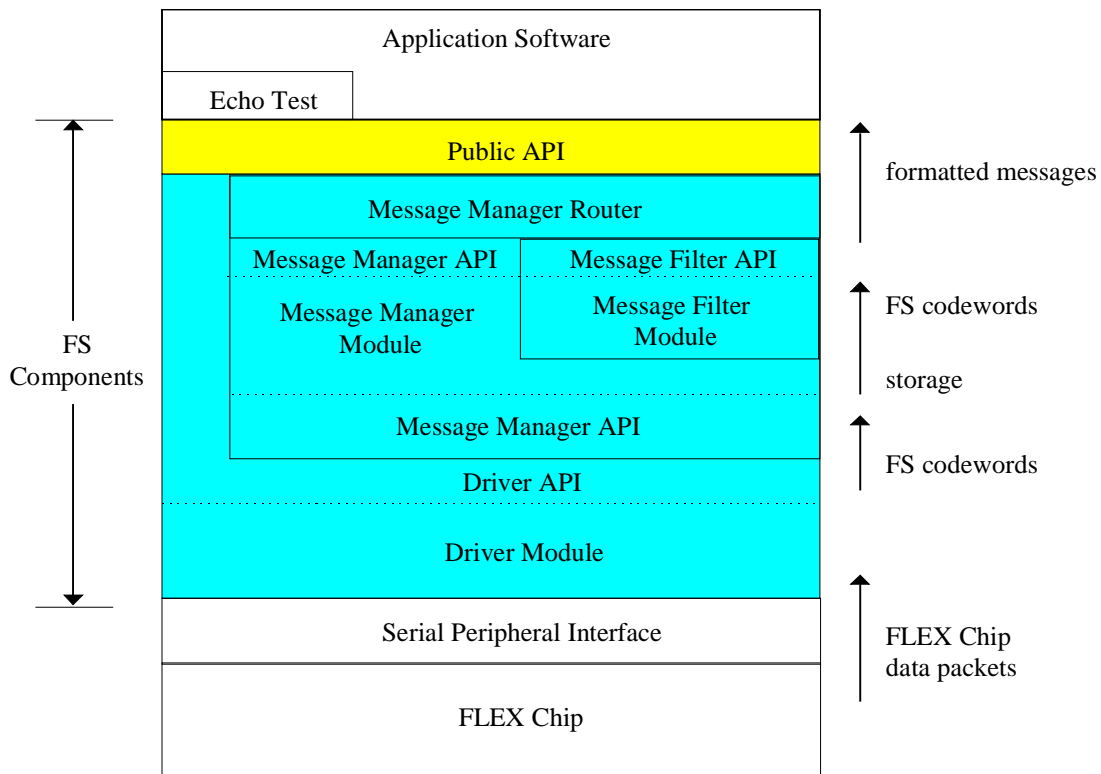
**Figure 1 - FLEX Frame Block Structure<sup>[1]</sup>**

### ***FLEX Stack Structure***

There are encoding and decoding rules which identify the minimum requirements that must be met by the paging device, paging terminal or other encoding equipment to properly format a FLEX data stream for RF transmission and to successfully decode it.

FLEX Stack One-Way (FS) simplifies incorporating a FLEX protocol into a wide variety of devices and appliances to process information received and demodulated from a FLEX decoder. FS is an Application Programming Interface (API) which runs on a host processor to manage communication with the FLEX decoder. FS handles the initialization, buffering of received code words, and decoding of separate address, vector and data packets. It also performs phase de-multiplexing, roaming, security, and event notification (out of range, low battery, time of day and error conditions).

The software consists of three modules and public and intermodule application program interfaces that control the FLEX decoder and manage the raw FLEX data as illustrated in Figure 2. The Driver module manages the flow of data from the FLEX decoder and builds raw message data from received data streams. The Message manager module stores and manages message data and routes calls to the appropriate API. The Message filter module formats raw message data according to the required format. This format could be ASCII characters, binary data, ideographic character symbols, or any other supported form. A Public Application Programming Interface (PAPI) complements the three fundamental FS modules. The PAPI provides a high-level interface that is used by the host software to manage message data, message notification, and the FLEX decoder using eleven function calls.



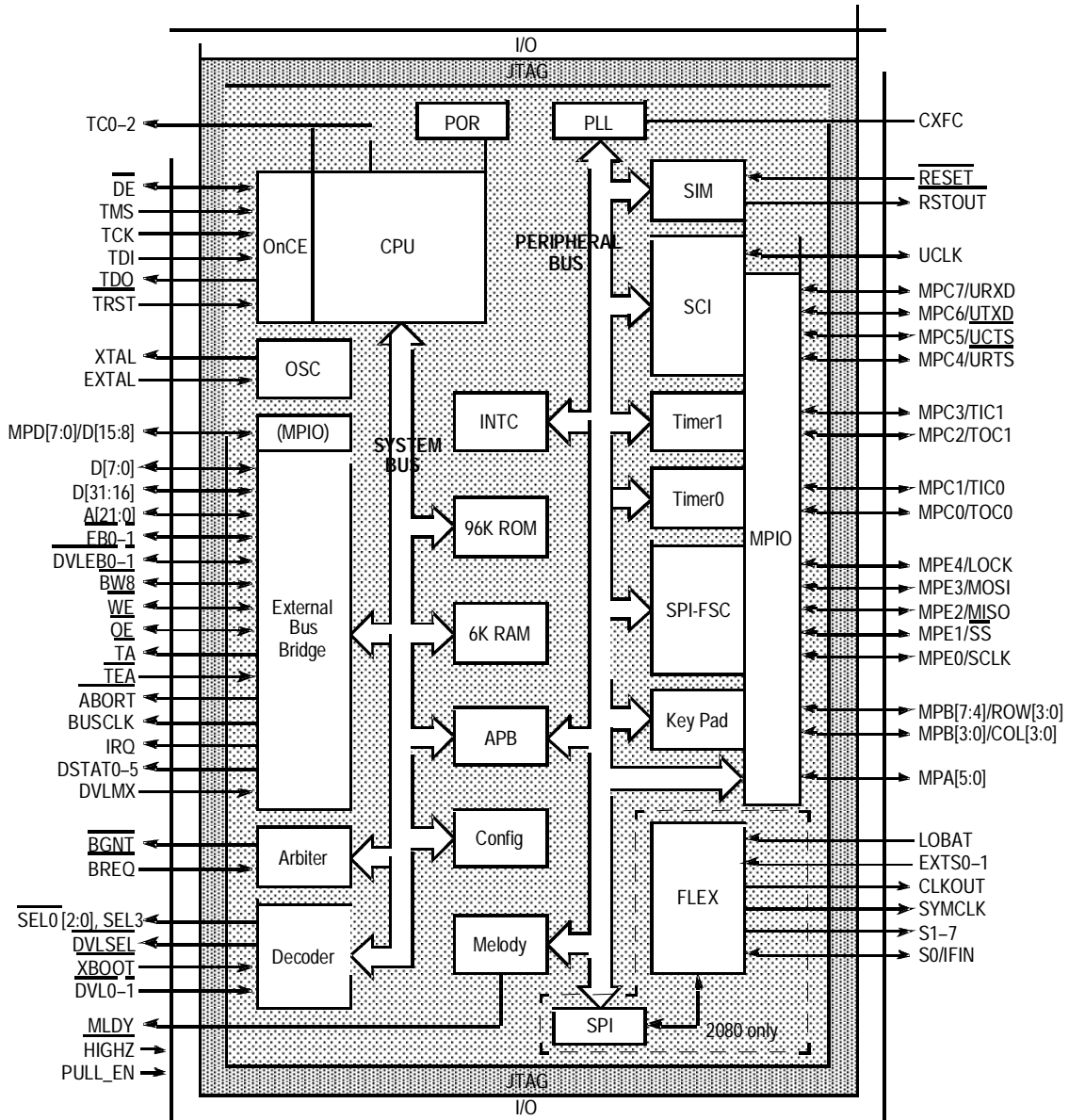
**Figure 2 - FLEX Stack One-Way Version 4.1<sup>[2]</sup>**

***MCM2080 for FLEX Roaming Paging Applications***

To facilitate FLEX application development, a system solution has been made available whereby the details of the protocol, hardware and software interface are made easier. The MMC2080 represents an integration of several field-proven technologies providing a versatile Roaming FLEX solution. This device combines the control and I/O capability of an M•CORE RISC processor with the processing power of the Roaming FLEX Alphanumeric Decoder core to provide a complete baseband solution for a paging or messaging system.

It features an integrated FLEX protocol decoder, digital demodulator, alert generator, and time-of-day timer. These features make this device ideal for mid-tier pagers and advanced messaging solutions. The FLEX decoder in the MMC2080 is the G1.9 compliant version for roaming, allowing paging carriers to network many paging systems across a broad geographic region and provide complete service to the customers who want nationwide service, all without disrupting service to existing local customers. This feature is important in countries like China and Korea that operate on different frequencies from region to region.

In addition to roaming capability, the FLEX G1.9-compliant decoder features partial addressing that increases battery life up to 25% and an internal demodulator that eliminates the need for a discrete analog-to-digital converter.



**Figure 3 – MMC2080 Block Diagram (144-Pin)<sup>[3]</sup>**

The MMC2080 architecture implements a low power M•CORE RISC processor with 24K x 32 bits of ROM (96K bytes) and 1.5K x 32 bits of RAM (6K bytes) as illustrated in Figure 3. The M•CORE processor is a streamlined execution engine that provides many of the same performance enhancements as mainstream RISC architectures. It is implemented with a fixed 16-bit instruction length and 32-bit internal data path that meets the

computational precision requirements of newer advanced products with the cost and power advantages previously available only with 16-bit architectures. Thus, increased code density accomplishes the goal of minimizing the overhead of memory system energy consumption.

To provide optimal static power management for the overall system, the M•CORE processor provides three instructions (stop, wait, and doze) that enable external logic to disable power to parts of the system. These instructions provide power management to internal peripherals as well by shutting down unused circuits that are not needed while waiting for the pager's particular frame.

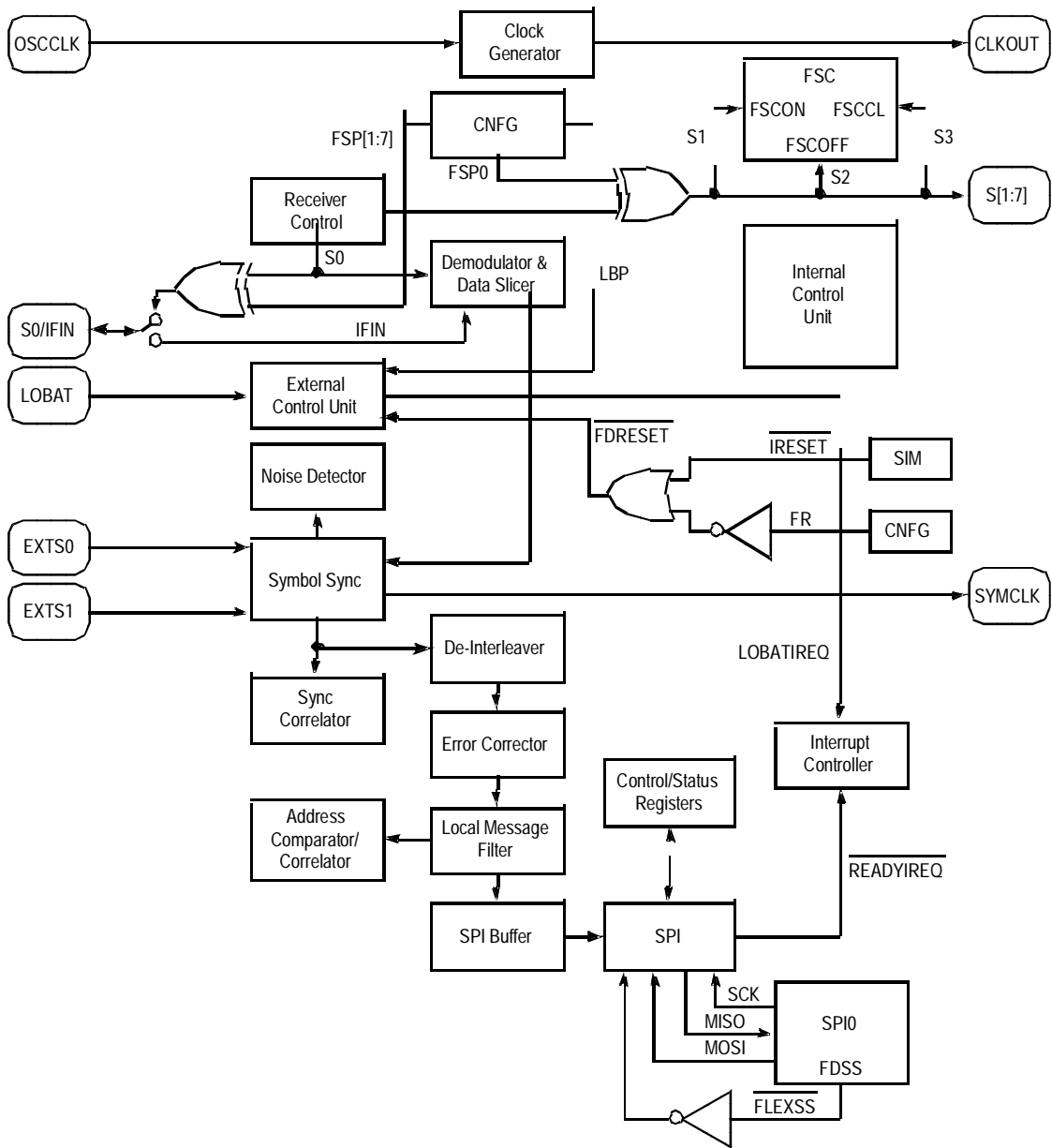
The M•CORE processor communicates through the APB Peripheral Bridge, which is the interface between the system bus and the peripheral bus. Operation of the APB is completely automatic and does not require any programming. The M•CORE processor and its associated peripherals have both DOZE and STOP low-power standby modes. Through the System Integration Module it is possible to place many of the individual modules on the MMC2080 into one of the low power modes independent of one another, thus providing the greatest possible flexibility in power saving.

### ***FLEX Decoder Module***

The FLEX decoder simplifies implementation of a FLEX paging device by interfacing with most industry-standard paging receivers. Its primary function is to process information received from a FLEX radio paging channel, select messages addressed to the paging device and communicate the message information to the resident M•CORE processor as illustrated in Figure 4.

The FLEX decoder supports 1600, 3200, and 6400 bps decoding. Intermediate frequency signals are demodulated, synchronized, de-interleaved and error corrected prior to entry into a holding buffer. The holding buffer is then fed into a synchronous serial peripheral interface (SPI) which then converts the data into a parallel format. All communication with the M•CORE processor is through the SPI control/status and data registers. The SPI is interrupt driven to reduce processor overhead and save power.

A resident real-time clock operates off a 76.8 kHz or 160 kHz crystal and is used to wake up the M•CORE processor at 1-minute intervals. This permits the M•CORE processor to operate in a low power mode when monitoring a single channel for message information. A low battery detect input is also available and generates interrupts to the M•CORE processor for display of a low battery symbol on a user interface and graceful shutdown.



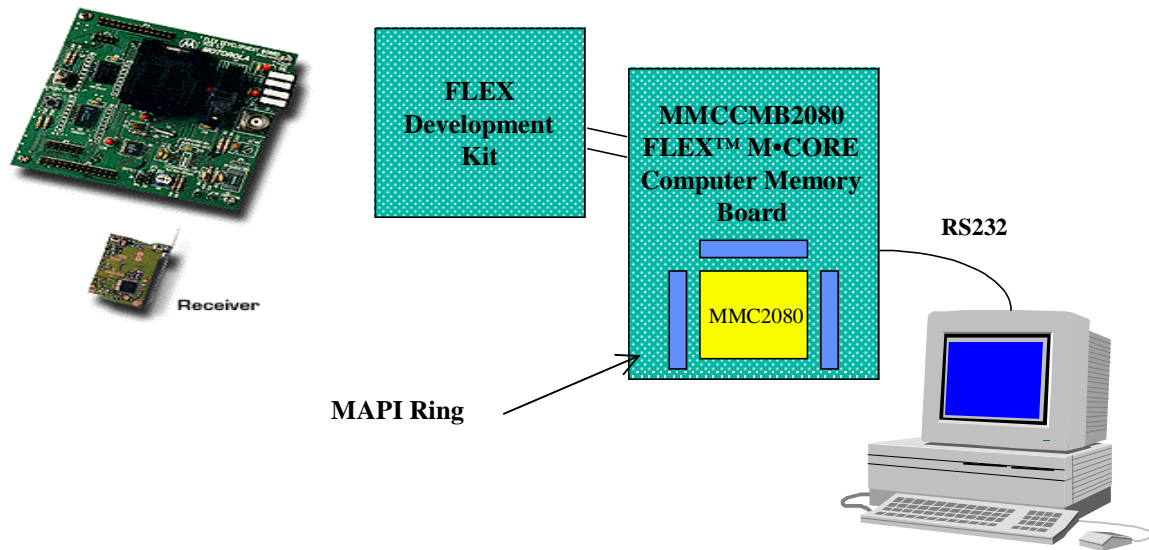
**Figure 4 – MMC2080 FLEX Decoder Block Diagram<sup>[4]</sup>**

### ***Developing FLEX Applications***

Developing products with messaging capabilities is now easier for the hardware and software designer. Development tools are available which facilitate rapid evaluation of the FLEX protocol. The FLEX Stack Roaming protocol software has recently been ported to the M•CORE processor and is available for licensing.

The MMCCMB2080 Computer Memory Board provides an easy interface to a RF receiver and a host computer for system level prototyping. The board provides a

Motorola development tools interconnect standard known as Modular All Purpose Interface (MAPI). This interface permits connecting FPGA development boards for custom circuit development. Figure 5 illustrates a FLEX Development Kit that is being made available.



***Figure 5 - MMC2080 FLEX Development Tools***

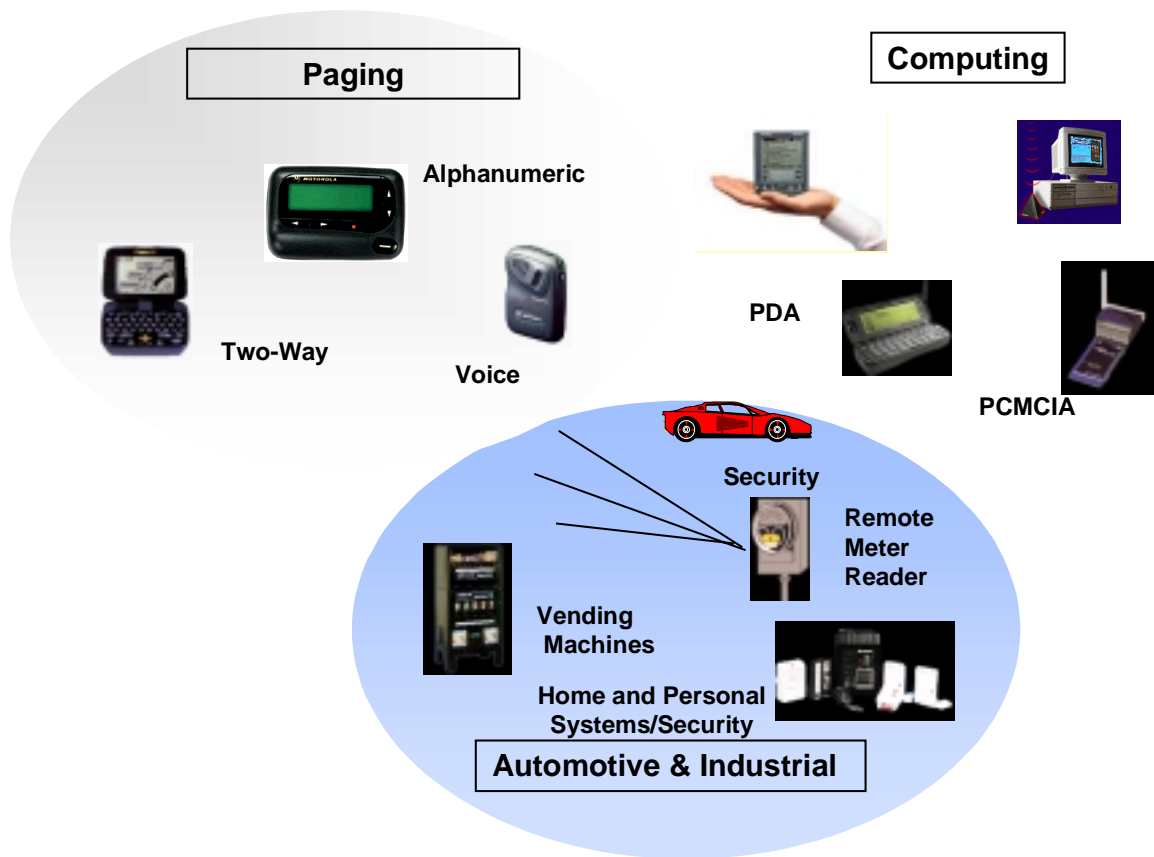
### ***ReFLEX™ Protocol – Two Way Messaging***

The underlying technology of the FLEX protocol allowed for the development of ReFLEX. A very brief summary of ReFLEX protocol will follow. The ReFLEX protocol provides an asymmetrical high capacity two-way data message delivery system for paging applications. It has a synchronous frame structure similar to and compatible with the FLEX protocol on a frame basis. It delivers control and data messages to subscriber units on a forward channel (outbound from the base transmitter) and for receiving acknowledgements and messages from subscriber units on a reverse channel (inbound from the subscriber unit to the base receiver).

This allows the paging device to “acknowledge” that a message has been received and also permits the service provider to monitor the paging devices’ location within a local geographic area using subscriber unit registration. Subscribers with keyboards are able to send back short response messages. Since the return channel is not coupled to the outbound paging channel, the message sent is independent of the “acknowledgement” that the paging device sends back as part of the protocol.<sup>[5]</sup>



ReFLEX systems are designed to operate on a frequency spectrum with a width that is a multiple of 25 kHz. A 25 kHz band supports a single digital FM control and data message channel, centered on the band. Digital FM channels must remain at a distance of 12.5 kHz from the edges of the available spectrum. A 50 KHz forward channel can support up to three digital FM control and data message channels separated by 12.5 kHz and operated in time lock. Outbound channels operate at 930-931 and 940-941 mHz range and inbound channels operate at 901-902 mHz. The protocol supports systems consisting of up to eight forward control channels. This opens an immense number of opportunities for low cost messaging applications as illustrated in Figure 6. The popular Motorola Page Writer™ 2000 two-way pager is an example of an implementation of the ReFLEX protocol.



**Figure 6 - Application Opportunities using Flex and ReFLEX protocols<sup>[6]</sup>**

## *Summary*

FLEX technologies are presenting a number of opportunities to improve our lives and help to produce new application areas that have not yet been tapped. There are now 229 operators in 48 countries in commercial operation or in process, representing 92% of the world's paging subscriber base. Japan, Korea, India, China and Russia have adopted the FLEX protocol as their national standard for high speed paging solution. This makes the FLEX protocol the global de facto standard for messaging.

New products are easier to implement when there is a complete solution to carry you from the conceptual stage to the delivery of production units. The integration of a FLEX decoder onto a low power M•CORE micro-RISC processor, the availability of the FLEX Stack software and a suite of development tools to accelerate product development makes the MMC2080 an attractive system solution for applications which require messaging capability. Device drivers are also available for each peripheral on the MMC2080 and will be offered in a software library. M•CORE architectures and FLEX technologies are strategic programs within Motorola.

Additional information on FLEX products may be obtained at the Motorola Website <http://www.mot.com/wireless-semi> or <http://www.mot.com/mcore>. Additional information on ReFLEX products may be obtained at the Motorola Messaging Website <http://www.mot.com/MIMS/MSPG/FLEX> .

### References:

- [1] The FLEX Family of Messaging Protocols, AMTTG145, Mar,1998, Motorola Advanced Messaging Technical Training Manual. <http://www.mot.com/MIMS/PSD/literature/literature.html>
- [2] MMC2080 User Manual, Appendix A, 1999, <http://www.mot.com/wireless-semi>
- [3] MMC2080 Product Brief, <http://www.mot.com/SPS/WIRELESS/products/MMC2080.html>
- [4] MMC2080 User Manual, Chapter 5, 1999, <http://www.mot.com/wireless-semi>
- [5] FLEX™ Technologies for Today and Tomorrow, Steve Torp, <http://www.mot.com/SPS/WIRELESS/information/flexarticle.html>
- [6] Special Thanks to John Wilson, MCU Applications & TDMA/Messaging Platform Manager, Wireless Communications Division/Motorola WSSG

FLEX, ReFLEX, M•CORE and PageWriter are trademarks of Motorola.