Texas Instruments DSP Solutions

Telephony Software Component Products

Overview

Today's advanced digital telecommunication products such as voice mail systems, PBX switches, and cellular MTSOs to name a few, require a level of programmability and throughput unprecedented in the history of telecommunications. Such computational throughput is needed for the soft implementation of functions such as speech / audio compression, DTMF detection and generation, and modems. The problem is amplified by the need to process multiple time slots on an T1 / E1 or ISDN interface. Programmable Digital Signal Processors (DSPs) provide this level of throughput along with level of flexibility and programmability unmatched with dedicated ICs or RISC-based solutions.

DSP Software Engineering, Inc. provides a wide range of telecommunication functions implemented on various fixed- and floating-point DSPs. These algorithms are highly optimized and structured, forming application building blocks which are easy to understand, integrate, and use.

Off-the-Shelf Products

DSPSE has qualified and tested a wide range of applications for the telecommunication market. Many of these software products are offered as re-entrant code (capable of processing multiple channels using only one instance of the code). This software is provided to our customers in a standardized C and / or assembly-callable form.

Fixed-Point Products (TMS320C25/'C5x)

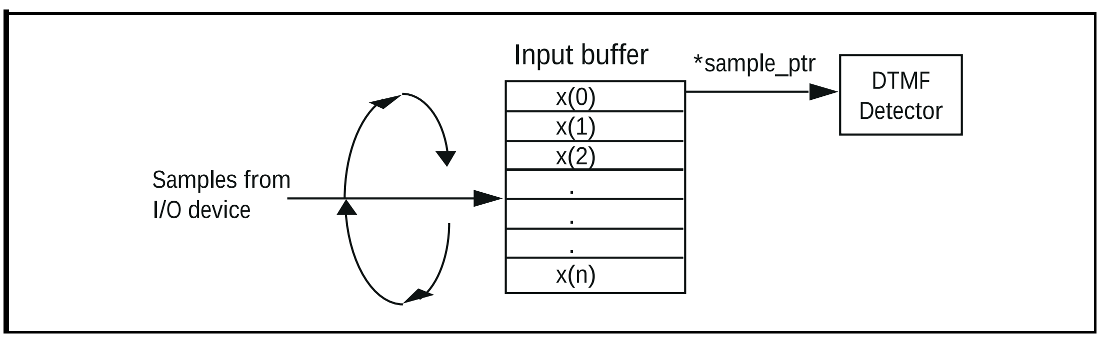
Description	Availability	MIPS	Is Code Re-Entrant?	Channels on 40 MIPS	Data Memory	Program Memory
Call Progress Detector	Now	2.2	No	N/A	45	512
DTMF Generator/Detector	Now	1.8	No	N/A	49	256
Caller ID Modem	Now	TBD	Yes	TBD	TBD	TBD
Line Echo Canceller (Adapt/Cancel)	Now	2.7/1.4	No	12/24	76/76	256
G.721, G.723, G.726 ITU ADPCM	Now	23	Yes	N/A	256	2.2 K

Floating-Point Products (TMS320C3x/'C4x Compatible)						
Description	**Availability**	**MIPS**	**Is Code Re-Entrant?**	**Channels on 20 MIPS**	**Data Memory**	**Program Memory**
Call Progress Generator/Detector	Now	0.52	Yes	36	1.0 K	1.1 K
DTMF Generator/Detector	Now	0.52	Yes	36	640	1.7 K
Caller ID Modem	Now	TBD	Yes	TBD	TBD	TBD
Line Echo Canceller (Adapt/Cancel)	Now	7.5/1.4	Yes	2/12	1.8 K	1.2 K
AGC and Voice Activity Detection	Now	0.2	Yes	24	125	650
G.711 μ-Law/A-Law Detector	Now	0.48	Yes	32	400	400
G.711 μ-Law/A-Law Compander	Now	0.8	Yes	24	0	300
Speech Time Scaler	Now	8.6	Yes	8 – 2	1.4 K	1.8 K
G.721, G.723, G.726 ITU ADPCM	Now	8.5 K	No	2	8.8 K	1.8 K

Integration Task

The integration of a DSPSE software component into the user's end system software is accomplished by simple application function calls like G728_encode(...) or G728_decode(...) followed by standard software build procedures (compile and link). System software for any platform consists of components such as boot code, processor-specific kernel (bus initialization, timers, interrupt handler infrastructure ...), I/O software, executive, and sometimes a command interpreter. What is actually written or implemented in the system software is a function of task complexity, system hardware, and I/O. The system software can be designed and implemented *by the user* or may be a full operating system licensed from a third-party vendor, such as SPOX from Spectron Microsystems. Operating systems are typically required for multiprocessor, multi-thread applications. DSPSE's software components "snap-in" to any system with minimal effort. All you need to know is the calling convention, naming conventions, system resource requirements (MIPs and memory), and the number of samples processed per task window period. Integration into a debugged system typically takes less than one day.

As noted, all of the telecommunication software components DSPSE licenses are provided in C and assembly-language-callable form. These components are application libraries which are "included" at link time. They are accessed by your system software with simple function calls. DSPSE writes these application libraries so that they are optimized for a specific processor and are host platform independent. Input data or sampled data is passed to the function via a pointer. This pointer, i.e., "*sample_ptr," typically points to an input arrays of samples and/or bits. This "black box" approach provides advanced DSP technology with minimal integration effort. Shown below is a graphic with some C code showing just how easy it is to "include" a DSPSE function.



```
int index;
int digits_buffer[MAX_NUM_DIGITS];        /* Buffer to hold detected digits */
obj_ptr = DTR_create(NULL, NULL);         /* Initialize DTMF Receiver       */
while (task_status == TASK_ACTIVE)        /* While task is active...        */
{
    Sample_ptr = AIC_get_frame(aic_object_ptr);
    if (num_digits_detected = DTR_detect(obj_ptr, sample_ptr, digits_buffer))
    {
        index = 0;
        while (num_digits_detected--)     /* Output digits to host          */
            HOST_transmit (digits_buffer[index]);
    }
}
```

Fixed Point or Floating Point?

Some of the major differences between a fixed- and floating-point DSP-based design are:

- Dynamic range of the number system (16- or 24-bit integer vs. 32-bit floating point)
- Accessible system memory and bus structure (width of address bus)
- Efficiency of the architecture to implement a DFT / FFT butterfly, FIR, or IIR filter structure
- The quality or efficiency of the compiler

All of these items affect the ultimate cost and performance of the system. This cost is a function of two variables, the unit cost of the DSP with the surrounding support logic and the size (word width length) of its memory. Historically, many designs have been developed with fixed-point solutions because of the low fixed-point per unit cost and 16- versus 32-bit memory width. However, the amount of engineering time and design skill needed to bring a multi-tasking fixed-point system to market could be two to three times that of a floating-point system.

Other considerations are fixed-point DSP restrictions such as memory reach and the lack of an efficient higher-level language. As a rule, an ANSI C compiler on a 32-bit floating-point processor will almost always outperform a 16-bit fixed-point machine compiler. However, for many systems, 64K of data and 64K of program space (16-bit address bus) is adequate if the software is developed solely in assembly language and the design is restricted to functions like DTMF decode / generation, call progress signaling, etc. If product requires fax, modem, speech prompting, speech recognition, speech compression, etc., then 64K of address space may significantly limit the system design. The following is desirable for a state-of-the-art PBX and / or central office digital switch:

- An extremely flexible architecture that can grow with demand
- Excellent HLL support
- Large memory space
- Multiprocessing environment

So, when should a fixed-point DSP be used in a telecommunication system design? In general, it is a function of system tasks, design experience, production volume, and time to market. Fixed-point systems make good sense for designs with over 5,000 systems per year, when a large and experienced design staff is available, and you can afford the additional time to market.

PBX and Central Office System Signal Processing

In a PBX and central office (CO) telecommunication system, the elements of the system can be categorized as follows:

- Handsets—*All devices that connect to the wall outlets at the customers' premise*

- Peripheral equipment—*The electronics that bridge the "switch" to the handsets*
- Common equipment—*The switch electronics that I/F to the PSTN network (E1/T1, DS0, DS1, etc.)*

Handsets

Handsets include a wide range of configurations and levels of functionality. There are currently two common forms of interconnect between a PBX/CO and a user handset:

- Analog loop—A tip/ring interface between the CO and the handset;
- Digital loop, called switch 56—This is a 56-kbps link through a CO between any two handsets within the PSTN

Examples of some state-of-the-art applications that DSPSE has implemented for various clients are:

- Secure phone systems
- Solid-state telephone answering machines
- Digital switch-56 data/voice phones
- Digital switch-56 wide-band phone systems (7.5-KHz audio band width)
- Digital switch-56 video phone

All of these applications have been implemented on either a fixed- or floating-point DSP. They use algorithms such as DTMF detection and generation, MRCELP, LDCELP, Caller ID, Speech Time Scalar, AGC, and Line Echo Canceller.

Peripheral Equipment

The peripheral equipment (PE) performs the interface between the handsets and the common equipment. Peripheral equipment predominantly consists of line cards. A line card can be designed to handle anywhere from 1 to 32 telephone lines (tip & ring). It performs a number of common telecommunication tasks like:

- DTMF tone detection and generation
- Echo cancellation
- Line equalization
- Automatic gain control
- V.32bis modem pool for system public interface
- Group 3 fax pool for system public interface
- Network interface for Switch 56 connect, ISDN interface, or X.25
- Voice compression and decompression
- Caller ID modem

DSPSE supports all of the above algorithms on either a fixed-point or floating-point platform.

Common Equipment

The Common Equipment (CE) performs switching and network control for the entire system. It interfaces directly to the PSTN over a variety of interfaces from DS0 to the X.25 packet-switched network interface. The CE is responsible for the connection of each line with any other service or line within the system. The specific switching requirement and method of implementation differs greatly between CEs. In some PBX designs, it is possible to combined the PE and CE within the same processor. In fact, this is common for equipment that interface to the PSTN via a signal T1/E1 line.

Contact: DSP Software Engineering, Inc.
175 Middlesex Turnpike
Bedford, MA 01730 USA
Tel: (617) 275-3733
Fax: (617) 275-4323
E-mail: sales@dspse.com