TEXAS INSTRUMENTS TCI PLATFORM: A COST EFFECTIVE, PROGRAMMABLE BASESTATION MODEM

Alan Gatherer (Texas Instruments, Dallas, Texas, USA; gatherer@ti.com)

ABSTRACT

The development of a hardware platform capable of supporting an SDR approach to WCDMA is a challenging design problem. In this paper we describe the approach taken in the design of the Texas Instruments Cellular Infrastructure (TCI) platform. The use of temporal domains to partition the problem along with loosely coupled functional accelerators allow a high degree of flexibility within the domain of CDMA modems as well as a standard DSP platform on which SDR techniques can be applied. We call this approach domain focused SDR.

1. INTRODUCTION

It is generally true that if a modem is initially developed using a mixture of hardware and software, it will not evolve back to a completely software solution. There are exceptions to this rule (for instance GSM handset modems [1]) but the reason it is generally true is that standards bodies tend to aim new technology at a level realizable by state of the art ASIC technology. Therefore all-software, real-time solutions are rarely available to intersect a new standard at its initial development. SDR has to catch up with the technology and this only occurs if the standard remains static for a sufficient period of time. This is shown conceptually in Figure 1. Note that as each new standard emerges the MIPs requirement jumps above what SDR technology is capable of and aligns itself closer to a mixed DSP/ASIC solution. How 'soft' the radio is can be seen as a balance between newly emerging standards and newly emerging SDR technology. Whether SDR can catch up depends on the length of time to the next standards evolution. The reader must be in no doubt though that the increasing complexity of standards and the shrinking time to market for these new standards means that modem manufacturers will endeavor to produce the softest radio that they can cost-effectively achieve.

In this paper we focus on implementation of the receive algorithms of a WCDMA modem on a flexible platform. A block diagram of top level functionality is shown in Figure 2.

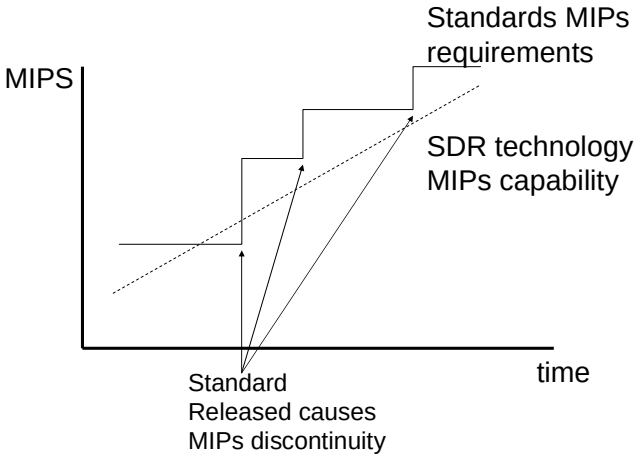

Figure 1: MIPs Requirements versus SDR Technology

We divide off the algorithm into 3 portions; that which cannot be done by a DSP without hardware assist (blocks marked with an A); that which is usually done in software (blocks marked with a C) and that which varies between SW and hardware assisted depending on the modem platform (blocks marked with a B). As the modem we describe performs the debated blocks in software the only blocks that require explanation are the A blocks. If we take, for example, the preamble detect task, this requires 100s of Giga correlations per second. With devices running in the 100s of MHz, such a task can only be performed if an adder tree (or equivalent bank of adders) is available to massively parallelize the correlation operation. This is generally true for the A blocks, and in the modem architecture we describe below, an adder tree is at the heart of the acceleration of the software. The trick is to develop an architecture that uses the adder tree efficiently while still maintaining flexibility within software. The blocks marked with A are described as performing chip rate processing because they operate on the fundamental samples of the baseband CDMA modem, called the chips. We will use one chip period as a measure of time in this paper. As the chip rate is 3.84MHz, one chip period is about 0.26us. Other WCDMA specific time periods used

in this paper are the slot (2560 chips or 666us) and the frame (15 slots or 10ms).

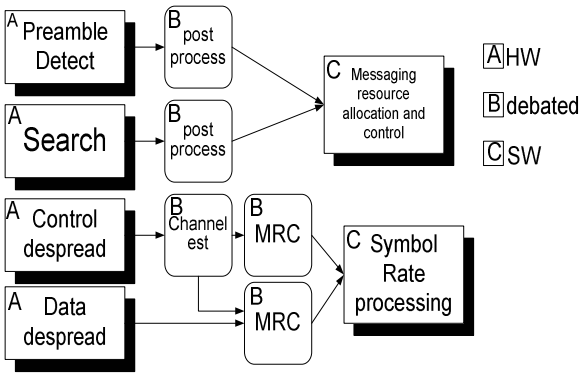

Figure 2: Simplified CDMA Modem Block Diagram

2. TCI RECEIVER OVERVIEW

In this paper we describe the Texas Instruments Communications Infrastructure (TCI) chipset. Given that we need to add acceleration for chip rate processing, the goal of the architecture is to be completely software-defined within the scope of the problem at hand, in this case CDMA. We call this "domain focused SDR". The resulting solution has enough processing power to also perform algorithms outside of the immediate CDMA-based application domain, such as GSM and EDGE. Thus the result is truly SDR within the domain of wireless basestation modems.

A block diagram of the TCI receiver is shown in Figure 3. We choose to loosely couple the correlation resources to the DSP. This allows the DSP to perform in parallel with the correlation operations. To maximize the flexibility of the solution we make the correlation accelerator as straightforward as possible. Essentially it is just a buffer, an adder tree and an accumulator. To minimize buffer overhead we attach the accelerator directly to a front end buffer from which data can be quickly extracted, processed and discarded.

The DSP controls the operation of the accelerator using task definitions. Each task, once set up, performs a simple operation continuously, dumping data into an interface buffer. An example is a single finger despread operation that takes data real time and produces symbols. It is then up to the DSP to retrieve this processed data and process it further.

The parallel operation of the two processing elements (DSP and accelerator) requires a third element to coordinate the transfer of data. This is conveniently provided by the DMA on the DSP. The DMA on the C6x TI DSP can perform a variety of function based on an internal list. The order of operations is therefore as follows

1. The DSP software requests a task be performed on the correlation accelerator with the results being dumped into a memory location in the DSP. It also may or may not request notification when the data is available.
2. A hardware abstraction layer on the DSP processes the request sending all the relevant information to the accelerator. The DSP may now perform other actions. The Host Interface on the accelerator is now primed to operate in unison with the DSP DMA to provide the correct transfer of data. It can perform this without intervention by the DSP.
3. The task is processed by the data path controller on the accelerator. The data sent by the DSP software contains information about when the task is to start or stop, so the task may lie dormant until the correct time. When the task is activated it starts generating data which the Host Interface will eventually transfer back to the DSP by writing to the DMA internal list the instructions given it by the hardware abstraction layer.
4. If required the DSP is notified by interrupt that data is available. The process for this new data is activated by the hardware abstraction layer using a software interrupt.

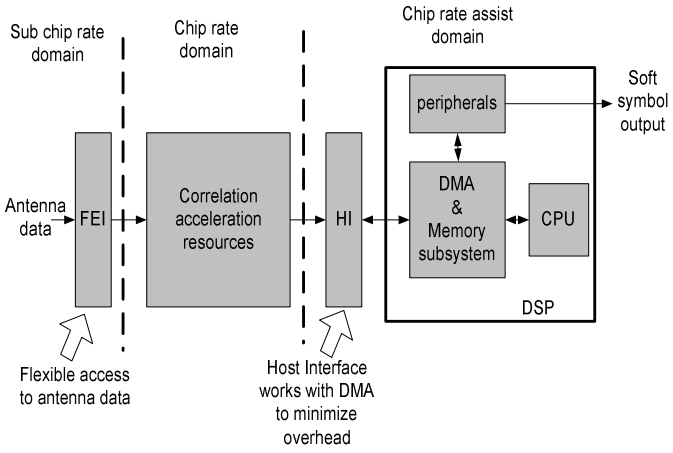

Figure 3: Basic Block Diagram of the TCI Platform

Notice that the correlation accelerator knows what time it is (i.e. it has a clock synchronized to the input data that is aware of slot and frame boundaries) but the DSP is only aware of the existence of real time. This allows the correlations to occur in real time as the data arrives and is rapidly discarded (to minimize the input buffer size). But the DSP need only meet deadlines, usually at the slot rate or above, and has no need to be synchronized at the chip level. For a 720MHz DSP like the TI C64x there are 48000 cycles

every slot so the timing requirements on the DSP are quite loose and a software solution becomes manageable. The DSP controls the chip rate accelerator to the chip level by sending clock information on when tasks need to be activated. New tasks are created fairly infrequently so control of the accelerator can coexist with the other functions on the DSP without onerous overhead due to interruption of longer processes.
The idea of separating tasks of different latency is discussed further in the next section.

3. TEMPORAL DOMAINS IN WCDMA

Development of a SDR modem is simplified by considering the different temporal domains of processing. An example of a temporal domain is the sub chip rate processing domain. In this domain the samples processed are coming in at a multiple of the 3.84MHz chip rate. Processing in this domain includes anything in front of the correlation operation and includes interpolation, buffering and subsampling. It is also important to note that in order to minimize buffering (and therefore cost) the processing loop tends to work on not more than a few 10s of chips of data. Table 1 illustrates a temporal domain partition of the WCDMA modem.

Temporal domain	Output Buffering requirement	Example functions
Sub chip rate	10s of chips	Interpolation
Chip rate	10s of symbols	Correlation, dispreading, energy accumulation
Chip rate assist	Slot or a few slots	MRC, channel estimation
Symbol rate	A frame	Imterleaving, TFCI decoding

Table 1: Temporal Domains

We wish to keep temporal domain on different processing elements, as much as possible. This prevents longer latency processes from being interrupted constantly by low latency processes and means that each processing element is only synchronized to real time as much as it has to be. This simplifies the programming of the device, as well as reducing OS overhead, because we are not mixing low latency and high latency processes. Though the DSP has control over the chip rate domain on the accelerators, the control is performed at the chip rate assist temporal domain and therefore coexists easily with chip rate assist domain processes.
On the TCI platform the sub chip rate and chip rate domains are processed on the accelerator. They are synchronized to a chip rate clock to minimize the input buffer overhead requirement. Small quantities of data are read in and all the active processes are run on this data while more data is being collected in a ping pong fashion. The processes are run serially on the data, the only requirement being that the total processing time is less than the time taken to collect the next buffer. Therefore a variable number of different tasks can be run on this accelerator. Each task will run for a few clock cycles each buffer. Because the list of active processes runs for the same time in the same order, the organization of the processes is straightforward.
The chip rate assist processes are run on a DSP. These processes have a much looser timing requirement and the DSP uses the operating system to abstract the processes away from their real time constraints. Each process will typically run for hundreds of clock cycles before completing. Processes will also be created here to deal with generation of commands to the accelerator. These are infrequent, occurring for example if there is a new user in the system.
The HI/DMA combination provides a transparent bridge between the chip rate and chip rate assist domains, transferring data without the intervention of the processors on the accelerator and the DSP. The information required for this transfer is logically stored in the HI when a new task is requested by the DSP.
The symbol rate processes work on even bigger blocks of data and are generally placed in another DSP.

4. REAL TIME PROCESSING ON A PROGRAMMABLE DEVICE

Sections 2 and 3 above have described the various domains of real-timeness encountered in a WCDMA modem – varying from sub-slot (120-130μs in 3GPP FDD), slot (667μs in 3GPP FDD), to frame (10ms) and multiple frames processing. In this section we focus upon the tightest (sub-slot) latency requirements and explore the impact of these on how soft the chip-rate solution can be. For this purpose we consider the downlink (DL) power control processing in 3GPP FDD as an example.
Figure 4 shows the processing time available for DL power control processing. In DL power control a slot is sent from the basestation and received by the handset. The handset aligns its transmission to the basestation with the received slot and sends power control information. After another delay the basestation receives this slot and then has a certain amount of time to decode the power control information before it transmits the next slot to the handset. Without getting into the definitions of all the fields in the slots we can see that the standard defines a set amount of time for processing the power control bit. This processing is therefore highly latency constrained. In a well designed modem (where the high MIPs operations have been taken care of) the capacity of the modem is constrained by the

ability of the software to turn around on a tight latency operation. DL power control is one example of such a situation in WCDMA.

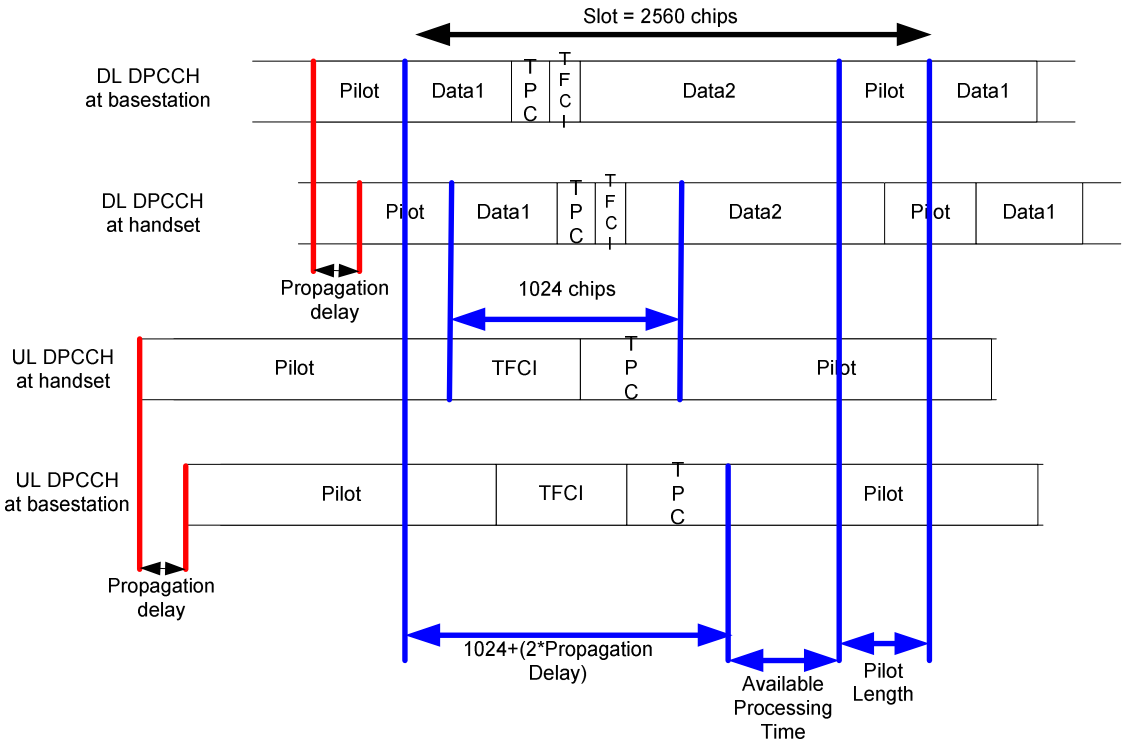

Figure 4 : DL power control latency requirement from 3GPP (TS 25.214) specifications

As seen in the figure, the available processing time is a function of two factors:

1. the DL DPCCH pilot length
2. whether power correction is applied to the slot immediately following or one slot later.

Let us take a typical case at spreading factor 128 (such as would be used for a voice call) and evaluate the available processing time budget.

For a typical voice user case in a 20km cell the worst case propagation delay will be 256 chips and the pilot length will be 256 chips. Hence

Available processing time = (2560 – (1024 + 2* 256) –256) *= 768 chips.*

5. HOW LATENCY DETERMINES THE CAPACITY OF THE MODEM

Let us now consider an implementation that is as soft as possible, thus providing flexibility of algorithm (such as channel estimation) implementation. The despreading, due to its highest operating frequency requirement, is performed in the accelerator followed by the demodulation of the bits and interpretation of the TPC command in DSP software. The latency chain for DL power control processing will now have 3 components, as shown in Figure **5**.

1. Latency through the chip-rate accelerator (say, this is 140 chips assuming an input buffer of size 64 chips and ping pong buffering),
2. Latency through the chip-rate DSP software (assume this to be 10 chips),
3. Latency due to communication with the transmitter LCCP followed by processing within the transmitter LCCP (assume this to be 110 chips).

This adds up to a total of 260 chips for a single user equipment (UE), which is well within the available time budget of 768 chips.

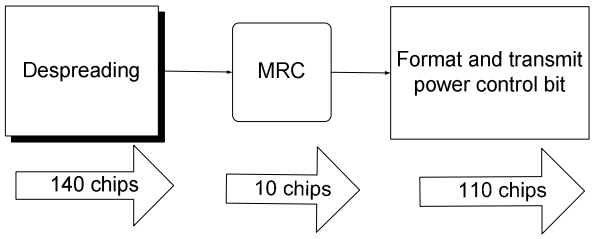

Figure 5: Example Latency in a Tight Latency Loop

This calculation is for only one user. In the worst case, all of the users will be slot boundary aligned so that they will need to meet the latency requirement at the same time. This will cause a spike in the MHz requirement that must be met by the DSP and will be the limiting factor on the number of users supported. This is shown conceptually in Figure 6. This is why the system is latency limited rather than MIPs limited. We now consider the case of 64 UEs on the TCI platform.

- The accelerator processes the data serially but is limited by the data input rate rather than the number of users (i.e. it is designed to process many users). So it does not add to the first latency in Figure 5.
- The DSP processes the data serially so the second latency must scale with the number of users.
- The format and transfer of the data is done using a peripheral and can be pipelined. So the only latency that matters is that of the last user to be processed.

In these circumstances, for the last UE to be processed in the DSP software, the total latency will be 140 + 64*10 + 110 = 890 chips, which exceeds the available time budget of 768 chips and implies the need to delay the application of the TPC command by an extra slot, at the cost of some performance loss. In this worst case situation, 51 users can be supported with zero extra delay on power control update. In practice this event is highly unlikely but it does allow us to conclude that the input buffering delay and the receive to transmit transport delay have to be kept to a few tens of chips (as is done in the TCI platform).

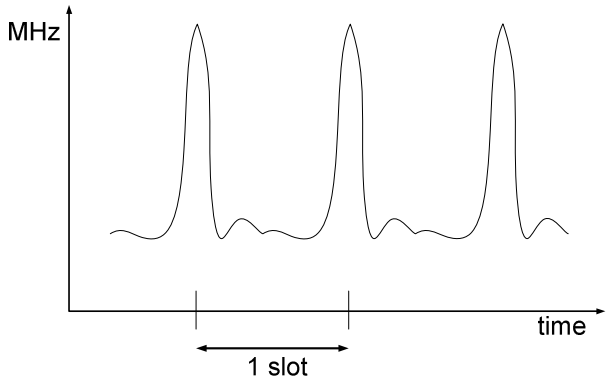

Figure 6: MHz with time in a worst case latency constrained system

Because, in the TCI platform, all other limiting factors have been minimized, the DSP peak MHz has become the main limiting factor determining user density. It is critical therefore that the overhead in processing the low latency processes (such as PC loops) is minimized. For instance, if a context switch occurs for every MRC in Figure 5, and each context switch takes one chip of time, the number of users supported with zero extra delay on power control will drop from 51 to 47. One chip corresponds to about 200 clocks on a 720MHz DSP which is large but could be used if the programmer is not careful. On the TCI platform the OS is configured to minimize the overhead due to switching in low latency processes and the effect of overhead on the channel density is minimal.

6. AVERAGING OUT THE MHZ

As described in the last section, the context switching and processing requirements of the low latency processes need to designed carefully. In the TCI platform the chip rate assist domain DSP runs these processes at highest priority and in a manner that flushes them through the system as quickly as possible. The other, higher latency processes on the DSP can then be processed at a lower priority. They will fill the "holes" in the MHz against time plot as shown in Figure 7. This does not contradict the philosophy of temporal domain separation from Section 3 because the rate at which processes are activated is the same. It is only the latency from receipt of data to completion of processing that differs.

6. SUMMARY

WCDMA cannot be done cost effectively completely in software. So an architecture has to be devised that supports an SDR modem implementation using accelerators. The TCI platform is such an architecture. It implements several important features, described in this paper, for an efficient SDR modem

1. separation of temporal domains to minimize context switching and simplify programming.
2. a simple adder tree based accelerator to take the heavy lifting out of chip rate domain processing.
3. task based control of chip rate processing from the DSP to avoid multiprocessor programming issues.
4. a smart data transfer mechanism to route data between the chip rate and chip rate assist domains without interrupting the DSP.
5. A two level priority system in the chip rate assist DSP to process low latency processes quickly and with minimum overhead.

With these features the TCI platform maximizes the number of channels supported by the DSP while keeping a single processor, SDR friendly programming environment with a high degree of user programmability. The only constraint on the software is that the front end data be processed by a dispreading operation. This is therefore a CDMA "domain focused SDR" platform. This architecture is also described in [2] and [3].

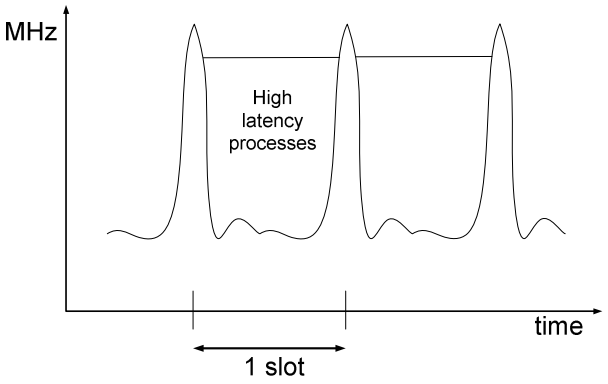

Figure 7: MHz with time in a worst case latency constrained system

7. SUMMARY

[1] A. Gatherer et al, "DSP-based Architectures for Mobile Communications: Past, Present and Future", *IEEE Comm. Mag.*, January 2000

[2] D. Hocevar et al, "Programmable DSPs for 3G Base Station Modems", chapter 4 in "The Application of Programmable DSPs in Mobile communications", Gatherer and Auslander, Wiley 2002.

[3] S. Sriram et al "A UMTS Baseband Receiver Chip for Infrastructure Applications", Hotchips 2003.