

Comparison of Processor Architectures for LTE Channel Estimation

Omer Anjum (Tampere University of Technology, Tampere, Finland; omer.anjum@tut.fi);
Teemu Pitkänen (Tampere University of Technology, Tampere, Finland;
teemu.pitkanen@tut.fi); Jari Nurmi (Tampere University of Technology, Tampere,
Finland; jari.nurmi@tut.fi)

ABSTRACT

In order to correctly demodulate the OFDM symbol it is very important to make a good estimate of the response of the channel and equalize the distortions caused to the transmitted signal. Channel estimation for slow fading channel has been implemented in this paper on different processor architectures for LTE with system bandwidth of 20 MHz. A comparison is presented in terms of programmability, flexibility, speed and power for the presented architectures. The compared architectures include RISC, DSP and Xentium (runtime reconfigurable design) which are the building blocks of state of the art SDR platforms presented by the industry and academia. In addition to these architectures Transport Triggered Architecture (TTA) is presented. Our study shows that TTA is one of the potential candidates for SDR (software defined radio) platforms which gives the designer much freedom to control various aspects of the whole system from software to hardware level.

1. INTRODUCTION

Most of the very high data rate broadcast applications today are based on multi-carrier techniques. The basic principle relies on the fact that high data rate stream is divided into multiple low rate data sub-streams. Each of these sub-streams is modulated on different sub-carriers which are all orthogonal to each other [1]. The main advantage of multi-carrier transmission is its reduced signal processing complexity by equalization in frequency domain and efficiency in frequency selective fading channels. Orthogonal Frequency Division Multiplexing (OFDM) proposed in [2] has been widely adopted as a very efficient multi-carrier digital modulation scheme to realize such systems. Due to very high spectral efficiency and immunity from ISI (inter symbol interference) of systems based on OFDM very high data rates, more than 100Mbps, can be achieved which is the primary goal of 3GPP LTE standard specification.

Achieving such high throughput, LTE demands very efficient implantation techniques at architectural level for

baseband processing. The processing block under consideration in this paper is channel estimation with maximum available system bandwidth of 20MHz according to standard specifications for LTE. In such system there are in total 1200 subcarriers for which estimations are to be made. The main problems in implementing such blocks with considerable demand for processing are strict timing requirement, power, flexibility and programmability. The latter two issues become significant to cope along with other issues when it comes to SDR.

Several different architectures from industry and academia e.g. RISC, DSP, Xentium and TTA have been considered for implementing channel estimation. A comparison has been made among those architectures in terms of the issues mentioned before. In addition attention has been drawn towards TTA architecture which holds a promising concept as a potential candidate for SDR platforms as compared to other architectures. TCE (TTA co-design environment) has also been discussed briefly as a very handy tool to develop TTA based platforms.

The rest of the paper is organized as follows. In section 2 LTE frame structure and resource allocation is described. In section 3 channel estimation problem is discussed. In section 4 implementation of channel estimation is discussed on different processor architectures. In section 5 results and evaluation is presented. Finally, section 6 summarizes the paper.

2. LTE FRAME STRUCTURE AND RESOURCE ALLOCATION

The physical layer specifications for LTE systems are provided in [3] and [4]. The transmitted downlink signal in LTE is defined by the resource grid which can be divided in time and frequency domain depending on certain system parameters. In general the frame structure of LTE systems, as shown in Fig. 1, consists of ten subframes, each of which is 1.0 ms duration. Each subframe is further divided into two slots of 0.5 ms each. Each slot can hold 7, 6 or 3 OFDM symbols depending on whether cyclic prefix is normal or extended.

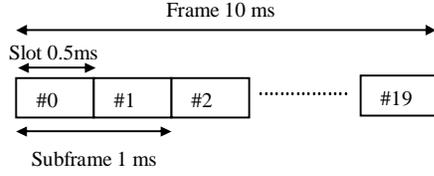


Figure 1: LTE frame structure

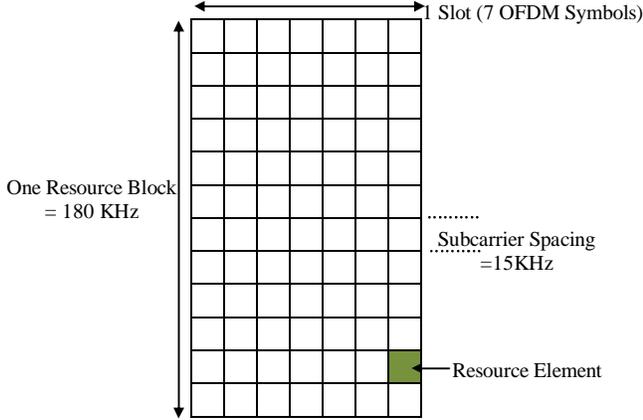


Figure 2: Resource Block structure of LTE

Depending on the bandwidth of the system each symbol contains the 6 to 100 resource blocks for 1.4 to 20 MHz system bandwidth respectively. The resource block is shown in Fig. 2. Each resource block groups 12 subcarriers with 15KHz subcarrier spacing. Each subcarrier is allocated to a resource element on which the used constellation symbol value is mapped [4].

3. LTE CHANNEL ESTIMATION

In order to correctly demodulate the OFDM symbol it is very important to make a good estimate of the response of the channel and equalize the distortions caused to the transmitted signal. OFDM based communication systems often make use of the reference signal named as preamble or pilot for channel estimation [5]. Depending on the channel characteristics (low/high frequency-dispersive channel, low/high Doppler channel or low/high frequency selective channel) there are different pilot configurations to equalize each subcarrier in OFDM based systems [6]. In block-type pattern of the pilot symbols channel estimation is based on different estimators like MMSE (minimum mean square error), Low-Rank Approximation, LS (Least square) estimator and reduced-order ML (Maximum Likelihood) estimator. MMSE and Low-Rank Approximation regard the channel as stationary random vector. So prior knowledge of the channel like the auto-covariance matrix and operating SNR is required which further increases the complexity. In MMSE matrix inversion is required for each symbol [5]. In Comb-Type

pilot symbols pattern we have time-domain windowing and frequency-domain interpolation. Time domain approaches need additional blocks for IDFT and FFT which further increases the complexity of the system. Channel estimation based on grid type pilot symbols pattern involves 2D MMSE interpolation which has a very high complexity and thus avoided in practical OFDM systems [5].

In this paper, a hexagonal grid of pilot symbols, subsampling the channel both in time and frequency domain, is considered. This pilot symbol pattern is useful when channel experience both time and frequency selective fading.

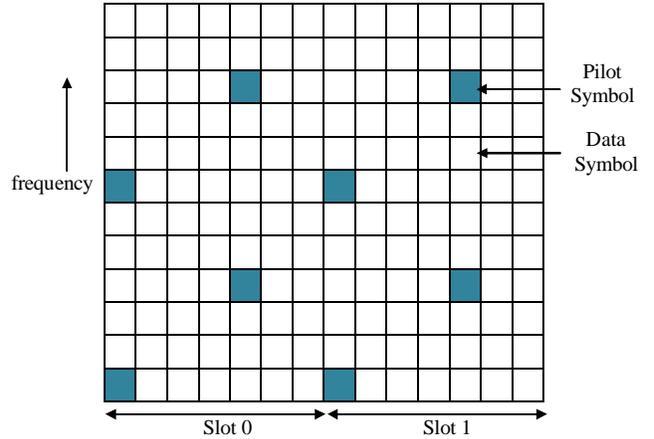


Figure 3: Hexagonal grid type pilot symbols configuration

The channel estimation problem can then be regarded as a two-dimensional problem, i.e., to estimate the channel both in time and frequency. However, as discussed above that due to the complexity of a 2D problem it can be broken into two one-dimensional problems. First estimate the channel in frequency domain and then later estimate the channel in time domain. In the evaluation, processing has been done for one subframe with a system bandwidth of 20 MHz which means that there are 100 RBs in one subframe and 1200 subcarriers in one OFDM symbol. It's the maximum system bandwidth of an LTE system according to the specifications. First very logical step in channel estimation is to divide the received pilot symbol by corresponding original pilot symbol to find out almost the exact response of channel on respective pilot locations. This estimate corresponds to the Least Square approximation.

$$H_p = Y_p / X_p$$

(1)

Where H_p , Y_p and X_p are the channel estimate, received pilot symbol and original pilot symbol. Channel estimate at all the other symbol locations still needs to be calculated. Using already calculated LS approximations channel estimate at all those remaining positions can then be interpolated using one of the available interpolation methods. In [16] different interpolation techniques have been compared for bit error rate on different values of SNR

(sign-to-noise-ratio). Linear interpolation method is a very simple approach with low complexity but the channel estimation quality can be improved further by using higher order polynomials. Their implementation complexity increases as the order of polynomial is increased. The one considered in this paper is piecewise cubic interpolation polynomial. According to [16] Spline Cubic Interpolation is the second best after the Low-Pass interpolation technique in terms of performance over a range of SNR values and almost same for middle and low SNR values. These techniques also provide a good trade-off between complexity and performance. Let us make some assumption that for every k -th subcarrier to be interpolated let

$$k/D = m + \mu$$

(2)

where,

$0 \leq \mu < 1$, D is the adjacent pilot symbol spacing for a subcarrier and m is the largest integer smaller than k/D . Then the piecewise cubic interpolation for k -th subcarrier can be written as follows:

$$\tilde{H}_k = C_1 \tilde{H}_{(m+1)D} + C_0 \tilde{H}_{mD} + C_{-1} \tilde{H}_{(m-1)D} + C_2 \tilde{H}_{(m+2)D} \quad (3)$$

where,

the coefficients of cubic interpolator are given as

$$\begin{cases} C_1 = -(1/3)\mu + (1/2)\mu^2 - (1/6)\mu^3 \\ C_0 = 1 - (1/2)\mu - \mu^2 + (1/2)\mu^3 \\ C_{-1} = \mu + (1/2)\mu^2 - (1/2)\mu^3 \\ C_2 = -(1/6)\mu + (1/6)\mu^3 \end{cases} \quad (4)$$

4. PROCESSOR ARCHITECTURES

In this section implementation of channel estimation algorithm, written in C language, for considered processor architectures is discussed. Each of the architectures execute one subframe containing two time slots, each with 7 symbols, and 100 resource blocks, and each with 12 subcarriers, as shown in Fig. 2. It is assumed that the architecture capable to complete this task in real time can cover the full range of LTE system bandwidths for channel estimation.

4.1. COFFEE (RISC Architecture)

COFFEE [7] is a general purpose embedded processor developed at Tampere University of Technology. Its architecture has adopted RISC philosophy, specifically targeted to support the compiler based software development for embedded systems. This core was developed with intention to work as general purpose processing node in a System-on-Chip environment or in a conventional embedded system for telecommunication and multimedia applications. It can also host a certain number

of coprocessors to accelerate application specific intensive tasks if needed to develop a suitable platform for a certain application. Its pipeline structure is shown in Fig. 4.

One subframe of LTE with system bandwidth of 20MHz was processed on a single COFFEE for channel estimation

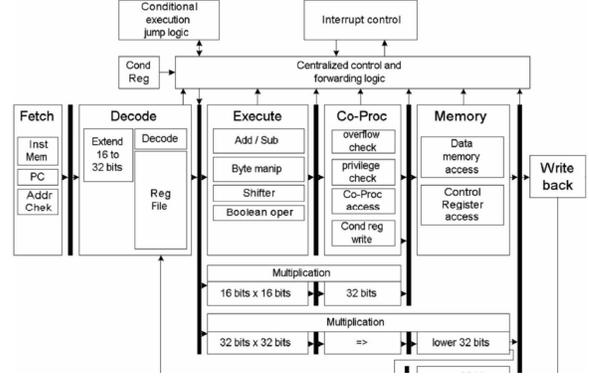


Figure 4: COFFEE Pipeline [7]

as described in the previous section. The task took almost 1,657,900 cycles to complete. To meet the real time constraint of 1 ms to finish the channel estimation task for one subframe, a single COFFEE needs much higher frequency than the one which has been achieved on different ASIC and FPGA technologies. In order to accelerate the task COFFEE was made to host an accelerator named as MILK. MILK can accelerate the division operation in our case which resulted in a reduced cycle count of almost 322,000 which in turn require around 322MHz operating frequency in order to meet the time limit of 1 ms to complete the task. Based on the published results in [8], achieving this frequency might not be possible on FPGA but on a modern ASIC it might be possible. COFFEE without MILK was synthesized on StratixIV and the dynamic power consumption using the switching activity simulation was found to be 741.75mW or 1.12 mJ/Task @ 181MHz.

4.2. Homogeneous MPSoC

The MPSoC used in this work is based on an NoC [9] developed at Tampere University of Technology. There are

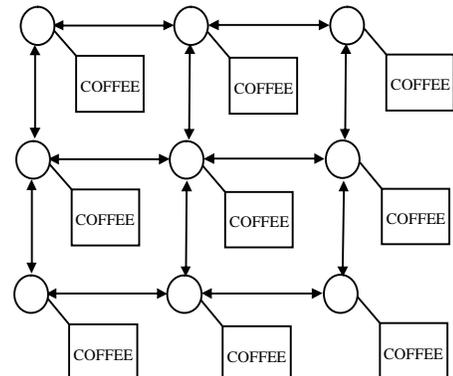


Figure 5: Nine NoC nodes hosting COFFEE RISC

in total 9 network nodes connected in a 3x3 mesh grid fashion, as shown in Fig. 5. Each node can host different processing elements which can communicate with each other using interface bridges. Bridges are then connected to global network switches through regular I/O links [9].

The task of channel estimation is distributed over nine COFFEE nodes for different set of subcarriers. All of the nodes work in parallel. Central node is the master node and remaining eight nodes being the slaves. Central node in addition to its own share of channel estimation load is also responsible for distributing the data to the surrounding eight slave nodes. The synchronization between master and the slaves is done by shared memory space which is accessible by all the nodes in the network. As the first step the master sets a flag, through a broadcast, in the shared space of each slave to indicate that it is ready to send the data. Slave in return, if ready to receive the data, writes to the shared space of the master node the relevant flag. The master node then feeds the data to all the nodes and signals them to start the processing. In the meantime, the master also does some portion of the iterations and waits for the completion flags from the slaves to go up. The task completion flag from the slaves is set only when the results from the slaves are returned to the master node. The data in our case was evenly distributed among all the nodes. The total number of cycles taken in this case were around 271,577, which is 6 times faster than that of a single COFFEE. To finish the task in 1 ms the system needs to run at about 272MHz. If we compare the speed/area ratio between the a single COFFEE with a MILK coprocessor and NoC the prior solution seems to be a better choice. NoC hosting COFFEE was synthesized on stratixIV which gives the max operating frequency of almost 180MHz and 1.033 mJ or 684.47 mW of dynamic power to complete the task. From the energy and power point of view, the multicore implementation is slightly better than the accelerated single core.

4.3. Xentium by Recoresystems

Xentium [10] is a fixed point VLIW-DSP optimized to perform digital baseband processing tasks. It consists of control logic, data path, instruction cache and tightly coupled data memories. The data path consists of 10 functional units that can operate in parallel. The tightly coupled data memories are organized as parallel memory banks in order to make sure the simultaneous multiple accesses through internal buses or external slave interface by multiple resources. Its common bus interface can also be extended to integrate Xentium with NoC.

The same task was executed on Xentium core and was able to complete in 495,725 cycles, which needs Xentium to run at about 469MHz to meet the need of 1 ms timing constraint. According to the published results in [11]

Xentium is currently running at 200 MHz and consumes 175 μ W/MHz when implemented in 90nm technology. According to these figures we can thus approximate that the energy consumption for our task will be around .086 mJ.

4.4. TI's TMS320C6416 DSP

TMS320C6416 [12] is TI's fixed point VLIW-DSP processor. It accommodates two independent data paths with four functional units (one multiplier and 3 ALUs) and 32 of 32-bit general purpose registers each. Data paths are also provided with a cross communication link to achieve better instruction and data level parallelism inherent in DSP applications. Its instruction set is also extended with TI's specialized instructions to achieve the accelerated performance in certain applications. LTE channel estimation task was run on this processor and completed in 403,692 cycles, which requires at least 404 MHz operating frequency to fulfill the timing requirement. From Table 1 in [12] it seems possible that it can achieve the real time system requirement running at 500MHz and at least consuming 200 mW or 0.161mJ to complete the task.

4.5. TTA (Transport Triggered Architecture)

Transport Triggered Architecture (TTA) is based on only one instruction called "MOVE" [13] where computations are done as soon as the data arrives on a triggering port of a functional unit. No particular instruction set architecture is defined for TTA. A typical architecture consists of several number of buses, functional units, register files and load store units. There are no restrictions how units are connected, it can be fully connected or point to point, or anything between them. In this way a TTA instruction consists of slots, one for each bus to specify the move operation on the respective bus in a clock cycle. So it more closely resembles a VLIW architecture. Main difference to VLIW is that all FU's must not be connected to all of the RF's (or even none) of the same cluster. This reduces the complexity of the IC, but adds width to the instruction word. [14]. Unlike other VLIW architectures scaling up TTA is much less complex because the functional units and interconnection network are completely independent of each other.

The TTA codesign environment (TCE) [15] has been chosen to generate the processor and machine code for LTE channel estimation written in C language. In TCE the architecture can be gradually built and programmed by the programmer according to the application needs. Programmer has the control of every aspect of the processor like the required number of buses, registers, functional units, bus interconnections etc. Trade-off between flexibility and performance can easily be translated by the

programmer by making the right choices for the required functional units, their granularity level, other supporting units and the interconnections among the units. Highly modular structure of TTA helps the designer to introduce programmer-specific customized functional unit which if added to the system can help to accelerate the application by multiple folds. In TCE these units can be written and tested for their performance using the support of high level language C without going to their hardware description, thus reducing the design cycle of the whole system. Thus it is easier to scale the architecture of TTA yet maintaining the benefits of VLIW. TCE tools makes it a promising TTA design environment for very divergent application requirements and most importantly bridging the gap between hardware and software engineers which has been a seriously grown concern.

The architecture in Fig. 6 has been considered in our test case consisting of only four buses, one ALU, one 32-bit multiplier, one divider, two load/store units and two 16x32-bit register files. To complete our task it took 449,736 cycles, consuming dynamic energy of 0.091mJ or 40.40mW running at 200MHz clock frequency. TCE design environment provides tool to wipe out connections and convert the fully connected version to a reduced connection one with specified percentage of increase, say 1%, in cycle count which might result in a further increase in the synthesized frequency of the system.

5. Results and Evaluation

In this section, evaluation has been made on the basis of figures obtained from different implementation in order to make certain conclusions. Comparisons based on the figures in table1 are not so straight as different processors have been tested on different technologies like FPGA or ASIC. It is due to the lack of resources available to test the systems on same technology. But still an overall idea that how different architectures behave for the test case can be deduced. From figures in table1 it can be seen that TTA is 3.7 times faster than a single COFFEE RISC core and comparable to Xentium as well. It was synthesized on 180nm ASIC technology and was able to run on 200MHz even when it is fully connected, Viewing its performance on 180nm it can be assumed that if it is synthesized on 90nm or 45nm technology it can easily approach the limit of 470MHz to meet the real time constraint. Also further reductions in power consumption are expected on a smaller sacle technology. To make a more clear comparison with TMS320C6416 its TTA equivalent was considered which took 1,478,139 cycles, which is almost 3.7 times more cycles than TMS320C6416. It is in fact due to the fact that TI's DSP uses some special instructions such as for division operation. The emulated functions used currently in TTA are not well written and can be improved and written

similarly to TI's libraries for exact 1:1 comparison between the two. This task has been left for future work but to make a more fair comparison division units with latency of 7 clock cycles was added to the TTA equivalent of TMS320C6416 to avoid emulated function for division. As a result the cycle count was reduced to 273,867 cycles, which is 1.5 times faster than TMS320C6416 and needs lower frequency to meet the real time constraint. It can also be seen in table1 that the TTA's energy consumption is about 1.76 times better than that of TMS320C6416 even TTA is synthesized on a bigger scale technology as compared to TI's DSP. Further TTA was scaled and yet another custom functional unit for square root with latency 8 was added to the starting architecture shown in Fig. 6. The task was able to be completed in 144,814 cycles, which needs only 150 MHz to achieve the real time constraint of 1 ms. This is the lowest achieved cycle count among the compared processor architectures. A system running at a low frequency might also help to reduce further the power consumption. As the TTA is fully connected it is flexible and not strictly application specific. As being part of an SDR platform it may also be used for some other tasks like equilization or linear filtering but these are not considered here and are left as future work.

Table 1: Summary of Channel Estimation implementation

Architecture	No. of cycles	Energy / Task	Technology
Single COFFEE	1,657,900	1.12 mJ	Stratix-IV
NoC with nine COFFEE nodes	271,577	1.033 mJ	Stratix-IV
Xentium	495,725	.086mJ (approximatly)	90nm CMOS
TMS320C6416	403,692	0.161mJ (at least)	130 nm CMOS
TTA almost equivalent to TMS320C6416	273,867		
TTA	449,736	0.091mJ	180 nm CMOS
TTA with Custom units	144,814		

6. Summary

In this paper a brief introduction was given about the higher throughput LTE standard and its frame structure. Different processor architectures were discussed and the same LTE channel estimation algorithm was implemented on all of them. Six times speedup up was gained by NoC with nine COFFEE nodes as compared to single COFFEE but at the cost of more area of the chip and still not reaching the real time constraint. Even in case of a single RISC core with MILK accelerator, the cycle count was

reduced as compared to plain COFFEE core but it needs an ASIC version to meet 1 ms timing constraint. Xentium core was also not able to meet this requirement if running at 200MHz. TMS320C6416 was able to meet the real time requirement but the architecture used for TTA is smaller and even scaling it and adding customized units is easier. Also TTA took the least number of cycles with customized units and needs a lower frequency to meet the real time needs which can also result in lower power consumption as compared to all other solutions. On the other hand the bus interconnection is fully connected and has the flexibility to run other DSP tasks as well. TTA thus can be considered as one of the potential candidates in SDR platforms.

7. REFERENCES

[1] Bingham J. A. C., "Multicarrier modulation for data transmission: an idea whose time has come," IEEE Communications Magazine, vol. 28, pp. 5–14, May 1990.

[2] Alard M. and Lassalle R., "Principles of modulation and channel coding for digital broadcasting for mobile receivers," European Broadcast Union Review, no. 224, pp. 47–69, Aug. 1987.

[3] 3GPP TS 36.201, "LTE Physical Layer – General Description", Technical Specification Group Radio Access Network (Release 8).

[4] 3GPP TS 36.211, "Physical Channels and Modulation", Technical Specification Group Radio Access Network (Release 8).

[5] Tzi-Dar Chiueh, Pei-Yun Tsai, OFDM Baseband Receiver Design for Wireless Communications. John Wiley and Sons, 2007.

[6] Meng-Han Hsieh and Che-Ho Wei, "Channel Estimation for OFDM Systems Based on Comb-Type

Pilot Arrangement in Frequency Selective Fading Channels", IEEE Transactions on Consumer Electronics, Vol.44, No.1, Page(s):217 – 225, February 1998

[7] Kylliäinen J, Nurmi J, Kuulusa M (2003) COFFEE – A Core for Free. In Proc. International Symposium on SoC, pp 17–22.

[8] C. Brunelli, F. Campi, C. Mucci, D. Rossi, T. Ahonen, J. Kylliäinen, F. Garzia, and J. Nurmi, "Design space exploration of an open-source, IP-reusable, scalable floating-point engine for embedded applications," Elsevier Journal of System Architecture, vol. 54, no. 12, 2008

[9] Ahonen, T.; Nurmi, J.; "Hierarchically Heterogeneous Network-on-Chip," EUROCON, 2007. International conference on "Computer as a Tool", EURCON 2007.pp 2580-2586.

[10] <http://www.recoresystems.com/technology/xentium-technology/> (last accessed date: 6.6.2011)

[11] <http://soc.cs.tut.fi/2010/Heysters10.pdf> (last accessed date: 6.6.2011)

[12] <http://focus.ti.com/lit/an/spra811c/spra811c.pdf> (last accessed date: 6.6.2011)

[13] H. Corporaal, H. Mulder. "MOVE: A framework for high-performance processor design". International Conference on Supercomputing. 1991; 692–701

[14] H. Corporaal. TTAs: Missing the ILP complexity wall. Journal of System Architecture, 45(1):949–973, 1999

[15] P. Jääskeläinen, V. Guzman, A. Cilio and J. Takala, "Codesign Toolset for Application-Specific Instruction-Set Processors", in Multimedia on Mobile Devices, San Jose, California, USA, Jan 2007, Proc. SPIE Vol. 6507, 65070

[16] Coleri, S., Ergen, M., Puri, A., and Bahai, A., "Channel Estimation Techniques Based on Pilot Arrangement in OFDM Systems," IEEE Transactions on Broadcasting, vol. 48, pp. 223–229, Sept. 2002

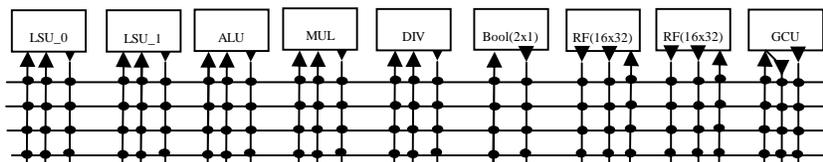


Figure 6: TTA architecture used in channel estimation