TxCP: A Coprocessor for LTE-A

Xin Huang^a, QingBin Liu, JunNing Wu, XueLiang Du and DongLin Wang

Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, 95 Zhongguancun East Road, 100190, Beijing, China

Abstract. With the widely use of 4G network, the corresponding bandwidth processing has become a critical issue. The current recognized 4G network is LTE-A. In the baseband processing for LTE-A, the processing of its physical layer algorithm is the biggest bottleneck for current processors. The use of application specific integrated circuit (ASIC) design has become necessary. This article will introduce a communication dedicated coprocessor (TxCP), specifically for LTE-A physical layer uplink shared/control channel (PUSCH/PUCCH) algorithm for bit-level acceleration. Its internal support for PUSCH/PUCCH CRC, Turbo encoding, equation definable convolutional encoding, data channel and control channel rate matching, channel interleaving, scrambling and modulation supporting QPSK, 16QAM and 64QAM. And in order to ensure that the coprocessor has a certain degree of flexibility, its internal controller design will support a variety of modes to ensure that some of the algorithm modules can be run separately. The programming model of the processor is relatively simple, and the user does not need to go through a complicated design.

1 Introduction

In the physical layer processing flow of LTE-A, the encoding process involves many small bitwidth bit-level processing, such as the characteristics of high density data operations (channel estimation), small bitwidth data interleaving characteristics (rate matching, channel interleaving), and some specific complex algorithms (Turbo decoding, Viterbi decoding). In different channels, these algorithms will have different application modes. The coding channels in LTE/LTE-A are also divided into various types, such as PUSCH, PUCCH, and PRACH [1]. In this article, we mainly support the PUSCH and PUCCH for coding and modulation process.

There is a high demand for peak rates in LTE physical layer solutions. LTE requires the downlink peak rate to reach 100Mbps, the uplink peak rate to reach 50Mbps [2]. This may be relatively easy for baseband processing designs. However, in LTE-A, the downlink peak rate requirements of 1Gbps, uplink peak rate requirements of 500Mbps [3]. This brings a lot of challenges to baseband processing.

At present, baseband processing is mainly based on general purpose processor (GPP), FPGA, digital signal processor (DSP) and coprocessor. But for the bit-level processing of the physical layer coding processing, GPP and DSP are difficult to achieve the desired speed requirements. For large-scale application solutions, the cost of FPGA is a problem, and FPGA will bring a lot of power problems. In this case ASIC solutions become a very good choice. In the article [4], we can see that ASICs have been specifically designed for hardware acceleration of PDSCH, PBCH and PMCH.

In this article, we present a coprocessor (TxCP) design specifically for PUSCH and PUCCH. The coprocessor specifically accelerates the encoding process for PUSCH and PUCCH. The entire acceleration process involves the cyclic redundancy check (CRC), turbo encoding, convolutional encoding, data information and control information rate matching, channel interleaving, and scrambling and modulation supporting for three modulation modes. In addition, in order to ensure that the operation modes of TxCP have some flexibility, TxCP internal parts of the algorithm modules can be separately scheduled.

The rest of this paper is organized as follows. Section 2 describes the overall architecture, internal modules, and the internal operations of the controller of TxCP. Section 3 introduces the results of the TxCP performance evaluation, including the synthesis results provided by Synopsys's DC tools. Finally, we summarize our work in section 4.

2 TxCP Architecture

The TxCP is internally composed of six configurable submodules for the LTE/LTE-A physical layer algorithm (CRC, TEC, CCE, RM, CIM, SMC). The configuration of each submodule is controlled by a unified top-level controller.

2.1 Overall Architecture

The TxCP is divided into three main parts: the controller, the memory, and the algorithm modules. The TxCP is

^a Xin Huang: huangxin2015@ia.ac.cn

internally controlled by a controller. The memory is responsible for storing the input data of the TxCP and the intermediate processing result of the algorithm modules. The algorithm modules accelerate PUSCH algorithms, a total of six modules. The overall architecture is shown in Figure 1.

Figure 1. TxCP overall architecture.

2.2 Cyclic Redundancy Check

The Cyclic Redundancy Check module completes the CRC check in 4G communication. The module consists of CRC-8, CRC-16, CRC-24a, CRC-24b four basic modules and control module. CRC-8 is mainly used for checking CQI information. CRC-16 is mainly used for downlink control channel and broadcast channel data transmission verification. CRC-24A and CRC-24B are mainly used for the verification of PUSCH data. CRC supports big/little-endian two different storage methods. And for multi-code block check, CRC supports both CRC-24A and CRC-24B. As the input interface is 8-bit width, CRC complements 0 in high bit for the code blocks which its bit length is not an integral multiple of 8-bit. And the check information is appended into the data information to output.

2.3 Turbo Encoding Coprocessor

The turbo encoding coprocessor (TEC) implements the turbo encoding of data information. In accordance with the requirements of the LTE-A protocol [5], this module has three-way information output, respectively, the original information, direct check information, and information interleaved in the memory and then encoded. The original information is essentially obtained by directly outputting the input data. The direct check information is obtained by encoding the input data. So the original information in this module and the direct check information is output in parallel. The third path information is obtained by interleaving all the data read into memory and then encoding the information after the interleaving is completed. This completes the 1/3 rate turbo coding function.

2.4 Convolutional Coding Engine

The convolutional coding engine (CCE) applies a more general solution. CCE can not only solve the LTE-A scenario, but also contain all the 1/4 bit rate, 1/3 rate and 1/2 rate of the constraint length of 7 convolutional coding, while supporting non-tail-biting and tail-biting convolutional coding. To meet these solutions, CCE allows the user to configure the equations required for convolutional coding through the registers in the TxCP. Therefore, in the LTE-A solution, only need to apply coding rate 1/3 to CCE. Figure 2 shows the convolutional coding solution in LTE-A with three-way convolution constraints of 133 (octal), 171 (octal), and 164 (octal). In order to achieve 1/3 rate, the fourth convolutional encoding of the configuration register is set to 0, the other three in its corresponding register can be configured in the corresponding value.

Figure 2. CCE configuration in LTE-A.

2.5 Rate Matching

The rate matching module supports the rate matching of the data channel and the control channel, and also has the function of code block concatenation. The rate matching module first needs to interleave the three sets of data streams from the turbo encoder or the convolutional encoder separately (the data channel and the control channel are interleaved differently) and are filled with null bits, and then the data is passed to the rate matching of the bit collection, selection and transmission module. The bit selection module is responsible for entering 3-way data streams into a circular buffer. The null bits are then deleted when selecting bits, and the result is output according to the start bit and length in the configuration [5].

Figure 3. RM module diagram

The above is an implementation scheme in LTE-A. In this design, a first-in first-out queue (FIFO) will be added between the interleaving matrix and the bit collector, as shown in Figure 3. This operation affects the order of bits entering the circular buffer. Therefore, in addition to receiving the three sets of data from FIFO after interleaved, the bit collection and selection module will receive empty bit location information to ensure the correctness of the data entering the circular buffer.

2.6 Channel Interleaving Module

Channel interleaving module (CIM), to achieve the RI, CQI, DATA, ACK four different information intertwined [6]. Channel interleaving module needs to support QPSK, 16QAM, 64QAM three different modulation methods, support different loop prefix mode, support different information of different interweaving. The module uses 6-bit bitwidth 10K memory as the physical mapping of the interleaving matrix, mapping the interleaved matrices of different number of columns (9,10,11,12) into memory. Using 8-bit output interface, the use of travel listed in the way, the output information to the lower module. For RI, ACK, and CQI information for some modules, cyclic fetches are made from 32-bit registers, reducing the dynamic power consumption of information in memory and reducing overall power consumption.

2.7 Scrambling and Modulation Coprocessor

The scrambling and modulation coprocessor (SMC) mainly scrambles and modulates the data outputted by the channel interleaver. SMC supports the generation of pseudo-random sequences, generating random sequences in a specific way according to the initial sequence set. And then the generated random sequence and input data XOR to scramble the data. Finally, the results of the scrambling are modulated. In this design, the modulation supports three modes, QPSK, 16QAM and 64QAM. In LTE-A, the placeholder X/Y will be used in the ACK and RI information. In the scrambling process, the position of the placeholder X/Y needs to be judged in order to obtain the correct symbol for modulation. So the mechanism used to determine the ACK and RI information is added to the SMC. The judgment process is based on the information provided by the CIM module.

2.8 TxCP Controller

The controller is responsible for the overall control within the TxCP, including the entire PUSCH channel encoding process and the control of the modulation process. TxCP supports up to seven operating modes through a finite state machine: Data only mode, Data + UCI mode, UCI only mode, CRC mode, TEC mode, CCE mode, CIM mode. The first three modes are the control modes for the PUSCH channel encoding process, which will schedule all the modules within the TxCP. The remaining four modes are for a single module within the TxCP.

Figure 4. The state machine transfer diagram of the TxCP internal controller in the Data + UCI mode

Take the Data + UCI mode as an example, as shown in Figure 4 for the TxCP controller internal state machine transfer diagram in this mode. The thick lines in the figure

represent the transition paths that need to be cycled in the case of multi-code blocks. In our design, TxCP supports a total of 13 code blocks, so the maximum number of loops is 13. In each loop, the RM is responsible for concatenation of the multi-code blocks. After the above cycle, the controller controls the TxCP to continue reading CQI information from the outside, via CRC, CCE, RM, and finally completing channel interleaving and scrambling and modulation, and eventually outputting the result.

There are 81 registers in the TxCP, 72 of which are configured for the internal algorithm module, and the rest is the overall configuration of the TxCP configuration. One of the registers is used to start TxCP. There is also a register that is responsible for storing the memory address where the data resides. After TxCP is started, the controller will determine the operating mode according to the contents of the configuration register, and reads the data into the internal memory of the TxCP according to the configured address. Then the controller will write the configuration registers of each submodule to the corresponding module. When multiple blocks are present, some modules have multiple sets of configurations. The controller of the TxCP will configure the algorithm module according to the number of blocks that should be handled at present. These internal registers of the TxCP are configured by writing to a Slave port of the TxCP via an external CPU or DSP.

After the algorithm module has been executed, TxCP will transfer the result data to the data output address specified by the configuration register. After doing the above, the controller sends an interrupt request to indicate that the task is completed. The CPU can write to one of the registers inside the TxCP to cancel the interrupt.

3 TxCP Performance evaluation

TxCP is designed to work at 600MHz frequency. So TxCP simulation test is also carried out in this frequency. In the current TxCP test, TxCP throughput performed well. Especially in the face of more data volume, TxCP can play the best performance. As shown in Table 1, it is the maximum throughput for the three operating modes for PUSCH. The throughput Thp_{in} is calculated as shown in equation (1), where D_{in} is the total input data and T is the total time from TxCP startup to sending interrupt request:

$$Thp_{in} = D_{in} / T \quad (1)$$

The output data in unit time Thp_{out} is calculated as shown in equation (2), where D_{out} is the total output data of the test case:

$$Thp_{out} = D_{out} / T \quad (2)$$

As shown in the table, TxCP has a high throughput in Data-only mode and Data + UCI mode. One of the main reasons why TxCP perform bad in UCI-only mode is that the input data in this mode is quite small, and the rate matching module and the channel interleaving module interleave and reuse the data multiple times, resulting in a large amount of output data. So the throughput in this mode will be very low. The corresponding Thpout has a

high value. We can conclude that the test case that produces the maximum throughput multiplexed the data so many times.

Table 1. TxCP Maximum throughput in different operating mode.

TxCP operating mode	Maximum Thp_{in} (Mbps)	The Thp_{out} of corresponding test cases (Mbps)
Data-only	297.44	1866.30
Data+UCI	289.64	1769.27
UCI-only	0.39	5128.78

Table 2. The synthesis results of TxCP.

Technology	TSMC 28nm
Area	25
Power	20
Frequency	19

TxCP not only has a high throughput, its area and power consumption on the same good performance. The TxCP synthesis is carried out under the DC tool provided by Synopsys. The results obtained are shown in Table 2.

4 Summary

As a solution for LTE-A PUSCH/PUCCH, TxCP has six modules for the communication physical layer algorithm. Part of the algorithm modules use a common design, to ensure that the TxCP has a certain degree of scalability and compatibility. The TxCP is controlled by an internal controller. The controller can support seven operating modes. There are three modes dedicated to the PUSCH of LTE-A, and the remaining four modes can individually schedule the sub-modules within TxCP to cope with more application scenarios. TxCP has a good performance, running at 600MHz under the maximum throughput of up to 297.44Mbps. TxCP also has a very good area and power performance.

Acknowledgments

This work is supported by the Strategic Priority Research Program of Chinese Academy of Sciences (under Grant XDA-06010402).

References

1. 3GPP TS36.211, Evolved Universal Terrestrial Radio Access (EUTRA); Physical Channels and Modulation (Release 11).
2. S. Sesia, I. Toufik, and M. Baker, LTE-The UMTS Long Term Evolution: From Theory to Practice, Wiley Publishing ,7 (6): 437-440 (2009).
3. A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, T. Thomas, LTE-advanced: next-generation wireless

broadband technology, Wireless Communications, IEEE, **17**(3) :10-22 (2010).

4. K. G. Digish, R. Thilagavathy, ASIC Implementation of Physical Downlink Shared Channel for LTE, ICCICCT, IEEE:370-376(2014).
5. 3GPP TS36.212, Evolved Universal Terrestrial Radio Access (EUTRA); Multiplexing and channel coding (Release 11).
6. FaTang Chen, Yan Chen, Research and DSP realization of uplink channel interleaver and deinterleaver in LTE system, AET, **37**(5):60-63. (2011).