VLSI Implementation of Rake Receiver for IS-95 CDMA Testbed using FPGA

Oliver Leung Chi-ying Tsui Roger S. Cheng

Department of Electrical and Electronic Engineering
Hong Kong University of Science and Technology
e-mail: cpegfa@ee.ust.hk e-mail: eetsui@ee.ust.hk e-mail: eecheng@ee.ust.hk

Abstract— In this work, an implementation of a time-multiplexed downlink Rake receiver complied with the IS-95 CDMA standard is presented. A low power architecture of the Rake receiver is implemented. A structure which provides the offset changing for the pseudo-random sequence (PN sequence) used for despreading of the CDMA signals is discussed. Architecture for the efficient time multiplexing of the Rake fingers is also presented. The design was implemented using Xilinx FPGA. It was tested to be functionally correct and the performance was complied with IS-95.

I. Introduction

Code-division Multiple Access (CDMA) has become very popular in wireless communication systems. IS-95 CDMA Testbed is a hardware testbed project to define and verify the baseband processing technologies, and try new baseband processing algorithm for the IS-95 CDMA mobile unit. As the power consumption in the handset has been one of the most important considerations in both system design and in implementation, another objective of this testbed project is to study the system tradeoff and come up with a DSP/ASIC partition in designing of CDMA baseband chip set for IS-95 mobile unit which has optimal performance and minimal power consumption. The whole base-band processor of the mobile unit was implemented in hardware and DSP. In CDMA spread spectrum system, Rake receiver [2] is used for the optimal demodulation for multipath propagation paths. The computational complexity and the number of Rake fingers required render it impractical to implement the Rake receiver in DSP code. In this paper, a design and implementation of a time-multiplexing version of Rake receiver using FPGA is presented. The overall structure of the demodulator is shown in figure 1. The focus of this paper is on the architecture which accommodate time-multiplexing for the rake fingers and the structure which achieve PN offset changing for configuration of the receiver. Also a low power Rake finger architecture is presented. In section II, the justification for using FPGA to implement the Rake receiver is given. In section III, the detail design and architecture of the Rake receiver are presented. The implementation details using FPGA are discussed in section IV.

II. Rationale of Using FPGA

During normal operation, each rake finger is required to obtain quantized data from the front-end of the receiver, doing IQ and walsh-code de-spreading as well as pilot averaging. These three tasks involve many additions and multiplications which are impractical to implement using DSP. Moreover, in order to capture multipath components, IS-95 specifies that three rake fingers, each of which responsible for demodulating one multipath component, have to be deployed. In addition, 8 time over-sampling is used in IS-95 to improve the performance of the receiver. In order to exploit the benefit of the over-sampling, early-late gate is implemented to facilitate the choosing of the correct sub-samples. This will need two more rake fingers for each demodulating finger. Thus, nine rake fingers in total are required for normal operation of the receiver. For additional features, such as fast acquisition, more rake fingers are needed. Therefore, we are targeting for totally 9-12 rake fingers in the whole receiver, which lead to a complexity that is unrealistic to be implemented in DSP codes. Therefore we decided to implement the rake receiver using FPGA.

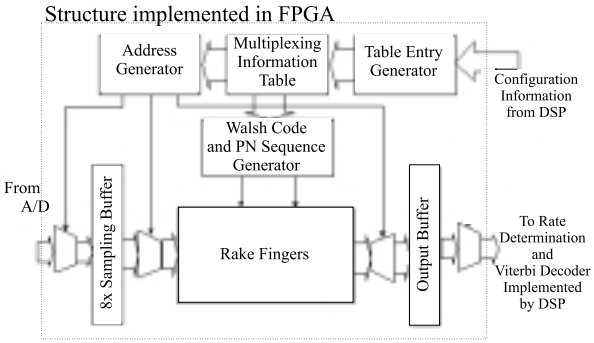

Fig. 1. Overall Structure of the Rake Receiver

III. Structure of the Rake Receiver

A. Structure of Rake Finger

In this implementation, we adopted the low power rake finger structure proposed in [1] which accommodate the pilot-aided coherent multipath demodulation used in IS-95. Figure 2 shows the detail structure of the rake finger described in [1]. In this structure, low power consumption is achieved by re-arrange the order of the three tasks performed by the rake finger(IQ de-spreading, pilot averaging and walsh-code de-spreading) from the conventional implementation [2]. Contrary to the conventional implementation of which the walsh de-spreading is done at the end of the Rake receiver, the Walsh de-spreading is moved ahead of the pilot averaging. In the conventional architecture, all the operations carried out before the de-spreading have to be run at the chip rate, which is 1.288MHz for IS-95. In the new architecture, since walsh de- spreading is done right after the IQ sequence de-spreading, the multiplication and addition process after the walsh de-spreading can be run at the symbol rate which is 19.2KHz for IS95. Thus, the average number of operation, and hence the power consumption is reduced substantially.

B. Multiplexing

In order to reduce the hardware overhead, we time-multiplex the rake fingers. Instead of having 12 physical rake fingers, our imple-

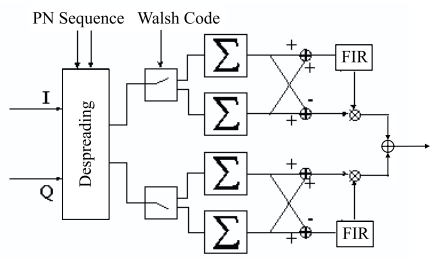

Fig. 2. Structure of the Rake Finger

mentation contains 3 physical rake fingers, each time-multiplexed 4 times.

To successfully de-spread the received signals, the correct PN and walsh code have to be supplied to each rake finger in every quarter of a chip period. These information are generated from the address generator, the pn-walsh-generator and the multiplexing information table(MIT). The MIT is used to store the configuration information such as the desired subchip position, register content for I and Q channel (explain in section C) and a counter storing the offset of the walsh code for each logical finger. In every quarter of a chip period, entries are read from the table (each entry corresponding to one finger). Normally, the address generator and the pn-walsh-generator will then use these information to generate the sub-sample address as well as the correct PN and walsh-code for the rake fingers. One exception is that if a logical finger is being reconfigured (indicated by the entry generator discussed in section C), the content from the table will be discarded and the one generated by the entry generator will be used instead. The entries in the table will be updated in every multiplexing cycle to keep track of the state (the desired sub-sample, and PN and walsh-code position) of each logical finger.

C. Entry Generator

Entry generator(EG) is responsible for generating the desired entries for the Multiplexing Information Table mentioned in section B. There is a master PN generator which is a 15 stage feedback shift register that acts as a reference point for the controlling DSP. When the controlling DSP wants to re-configure a logical rake finger, it presents the corresponding finger number, the desired subchip position and a 15-bit offset with respect to the master PN generator to the entry generator. In IS-95, the PN sequence is generated using a 15-stage feedback shift register with different feedback logic for I and Q channel. The content of the shift register (register state) for each rake finger is stored in the MIT and the PN code is generated by advancing the register states by 1. Therefore, the goal of the entry generator is to use the 15-bit offset to generate the correct register content as well as the Walsh code offset which are the required information for generating the PN and walsh-code for each rake finger.

Register content for both I and Q channel is generated by using pre-calculated matrix, V_n. Let $\vec{r}$(i) be a vector representing the content of a 15-bit register at certain offset i. Then, the values of the matrix are calculated so that $\vec{r}(i+2^n) = \vec{r}(i) \cdot V_n$. That is, giving a 15 bit register content to a matrix V_n, it will produce a new register content which is 2^n states later than the input. (we can see that the pn generator mentioned in section B is essentially equal to V_0) The structure of the PN generation matrix of the entry generator is shown in figure 3. In order to reduce the hardware overhead, we only implement the matrix $V_0, V_2, ..., V_{14}$ and use 3 cycles to generate the correct information. The 15-bit offset given by the DSP acts as the select bits for EG to determine whether to by-pass certain matrix or not and the register content of the master PN generator is the input to V_0. In the first cycle, the even bits of the offset are used as the select bits. In the second and third cycles, the odd bits are used. Therefore, the register content of any offset value between 0 and 2^{15} can be obtained in three cycles.

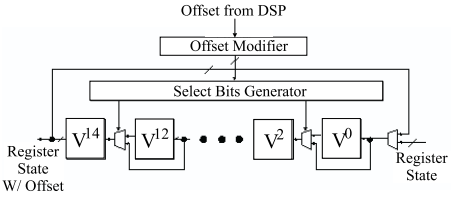

Fig. 3. Structure of PN Generation Matrix

Consider an example for a given 15-bit offset $[b_{14}, b_{13}, ..., b_1, b_0]$=[110000000001001]=$2^{14}+2^{13}+2^3+2^0$, let $\vec{r}$(i) be the register content of the master PN generator at arbitrary offset i. In the 1^{st} cycle, the even bits $[b_{14}, b_{12}, ..., b_2, b_0]$=[10000001] will select the path for $\vec{r}$(i) to go through V_{14} and V_0 only, i.e. $\vec{r}(i) \cdot V_{14} \cdot V_0 = \vec{r}(i+2^{14}+2^0)$. In the 2^{nd} and the 3^{rd} cycles, the odd bits $[b_{13}, b_{11}, ..., b_3, b_1]$=[1000010] make the result from the 1^{st} cycle to go through V_{12} and V_2 for 2 times, i.e. $\vec{r}(i+2^{14}+2^0) \cdot V_{12} \cdot V_2 \cdot V_{12} \cdot V_2 = \vec{r}(i+2^{14}+2^0+(2^{12}+2^2)\times 2) = \vec{r}(i+2^{14}+2^{13}+2^3+2^0)$.

In our implementation, the pre-calculated matrix operates at a rate of 4 times chip rate, which means one entry in the MIT can be changed per chip period.

One problem of this 'pre-calculated matrix' strategy is that it does not handle the 'additional zero' problem which is described as follows: The PN sequence used in IS-95 has a period of 2^{15} chips. However, if we generate the sequence using a 15-bit shift register, we can only have $2^{15}-1$ states because the 'all-zero' state will never happen. So, an additional zero needs to be generated after 14 consecutive zeros are seen from the output of the shift register.

To solve this problem in our design, there is an offset modifier before the given offset is fed into the matrix. The function of this unit is to reduce the offset by one if the current position of the master PN generator plus the given offset exceeds 2^{15} (i.e. cross the position of the additional zero). Thus, the register content generated using this reduced offset will be one chip earlier as if the 14^{th} zero is repeated once(act as the additional zero).

IV. Result

A time-multiplexed version of rake receiver using pre-calculated matrix for offset change was implemented using two Xilinx XC4028XL with utilization around 65% for each FPGA. The design was coded in VHDL and gate-level design was synthesized using Synopsys. The design was then mapped to Xilinx FPGAs. The receiver was tested at a clock rate of 10MHz and was functionally correct. Figure 4 shows the test board with the FPGAs on it.

References

[1] C.Y. Tsui, Roger S Cheng, Curtis C. Ling , "Low Power Rake Receiver and Viterbi Decoder Design for CDMA Applications," to appear in *International Journal on Wireless Personal Communications*.

[2] J. Viterbi, CDMA, *Principles of Spread Spectrum Communication*, Addison-Wesley, 1995.

Fig. 4. PCB with FPGA of the proposed design