ALL-digital AGC in CDMA Base Station Receiver

Qiang Du, Meilong Jiang, Guangquan Lin, Ning Sun
Eastern Communications Tech. Development Co.,Ltd
Email: duq@mail.eastcomtd.com

***Abstract*—This paper discusses two basic structures of AGC (Automatic Gain Control) and their corresponding hardware implementations in FPGA (Field Programmable Gate Array). Some area-efficient algorithms are developed and adopted; performance comparison between the two implementations through fixed-point simulation of MATLAB is presented.**

Keywords*—*CDMA; AGC; Feed Forward; Feed Backward

I. Introduction

In CDMA wireless communication system, the variation of signal level in the reverse link at antenna input might be more than 73dB[1] due to multi-path and fading. Traditional ADC (Analog to Digital Converter) does not have such a wide dynamic range, so AGC (Automatic Gain Control) must be added in front of ADC. On the other hand, it must be guaranteed that the complex digital power level of received signal to be processed in baseband maintain relatively stable, which makes AGC necessary in CDMA (Code Division Multiple Access) base station receiver. A traditional solution of AGC is implemented with a feedback loop and in an analog-digital mixed way, which consists of a power estimator, a DAC (Digital to Analog Converter), an analog op-amp integrator, and gain controlled amplifier in front of ADC, but the design costs more and has a high degree of imprecision. Recent advances in IC devices such as wide dynamic range ADC support direct IF (Intermediate Frequency) sampling and an all-digital AGC in wireless communication is then made practicable.

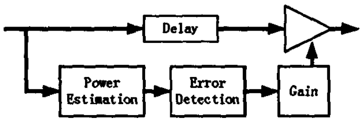

Figure 1. Conceptual structure of FFAGC

An all-digital AGC can be implemented in two basic ways: feed forward (FF) and feed backward (FB).

Conceptually a FFAGC is composed of three units: error detection, output scaling and time delay, while a FBAGC may contain four basic operations: error detection, loop filter, time delay and output scaling. As shown in Fig.1 and 2, the error detection part in both structures includes power estimation and error detector.

In CDMA2000 receiver, it is required that the complex digital power level P of received signal at the input of baseband processor be relatively stable:

$$P = I^2 + Q^2 = 18LSB^{[2]} \qquad (1)$$

In FFAGC, signal power is estimated at the input, and then the estimate is subtracted from a set point value to get the error. The error is converted into a corresponding gain value to adjust the delayed input samples so that the deviation of power level can be compensated.

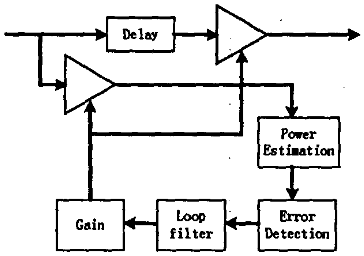

Figure 2. Conceptual structure of FBAGC

In FBAGC, power estimation is made at the output. The error between the estimate and the set point is calculated, and then the error is scaled and integrated through the low pass filter. The output of the low pass filter is the control word for the forward path gain. If the forward path output is larger than the set point, the error is negative, and the gain is reduced; if the forward path output is smaller than the set point, the error is positive and the gain is increased.

II. Implementation:

Each of the all-digital AGCs mentioned above is implemented respectively in a Xilinx Spartan II XC2S-150 FPGA in the design. Both have dynamic range of 52dB, adjustment resolution of +/-0.4 dB, and AGC time constant of about 200us.

A. Feed forward

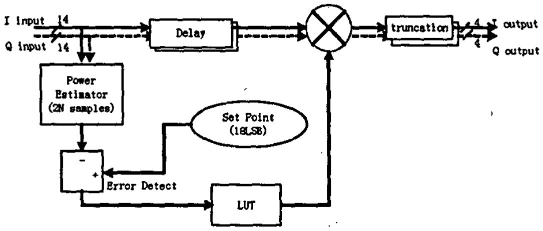

Figure 3. Practical structure of FFAGC

Fig.3 shows the practical structure of the FFAGC in FPGA. The power estimator calculates the power estimate of M pairs of I/Q samples. With every pair of samples sliding into the AGC, the estimate is made and a corresponding gain is produced. In order to simplify the hardware calculation algorithms in FPGA and increase the speed of AGC response, LUT (Look Up Table) is used to make the conversion from the detected error to the forward path gain. The depth of the LUT is settled according to the AGC dynamic range (R) and the power level resolution (r) of the output. Given the address space of the LUT is divided into L segments conterminously,

$$L = R/r. \tag{2}$$

The delay unit delays the input data of $M/2$ pair's sampling periods. The multiplier is multiplexed in time in both I and Q paths for space efficiency consideration. The AGC scaled outputs are truncated to required 4-bit width.

Since the power estimation is made base on 14-bit data, some specific algorithms are adopted in the implementation to reduce the resource occupation in FPGA.

In power estimation module, the average power level of M pairs of I/Q samples is acquired by

$$\widetilde{P}_s = \frac{1}{M}\sum_{M}(I^2 + Q^2). \tag{3}$$

It's a sliding window for averaging, in which every M pair of data for each sampling clock has to be registered. Thus it will be area killing if the averaging length is rather big in order to get an estimate accurate enough. In fact, suppose the power level is relatively stable, a two-step algorithm can be applied. Given

$$\widetilde{P}_1 = \frac{1}{M_1}\sum_{M_1}(I^2 + Q^2), \tag{4}$$

$$\widetilde{P}_2 = \frac{1}{M_2}\sum_{M_2}\widetilde{P}_1, \tag{5}$$

where $\widetilde{P}_1$ is the average of M_1 I/Q data, $\widetilde{P}_2$ is the M_2 averaging of $\widetilde{P}_1$, and

$$M = M_1 \times M_2. \tag{6}$$

If M_1 and M_2 are large enough, at time m,

$$\widetilde{P}_1(m) \approx \widetilde{P}_2(m-1), \tag{7}$$

$$\widetilde{P}_2(m) \approx \frac{1}{M_2}[\widetilde{P}_1(m) + \sum_{i=1}^{M_2}\widetilde{P}_1 - \widetilde{P}_2(m-1)]. \tag{8}$$

As seen in Fig.4, with each $\widetilde{P}_1$ sliding into (adding into) and a previous time's estimate $\widetilde{P}_2$ sliding out (subtracted) from the estimator, the estimate of $\widetilde{P}_2$ approximately equals to M_2 averaging of $\widetilde{P}_1$. Here only M_1 +3 data, which is much smaller than M, need to be registered, and it is thereby quite area efficient.

On the other hand, if $\widetilde{P}$ is estimated according to the operation mentioned above, the estimate would be 27-bit wide. And if it is applied as the input of the LUT, there should be at least L comparators of 27 bit, which would cost large amount of SLICEs in Xilinix FPGA.

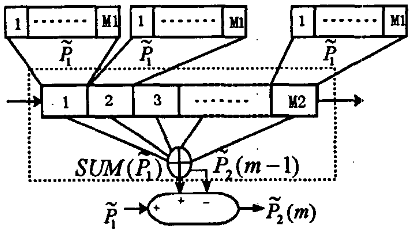

Figure 4. Structure of power estimation

In this implementation, the 13-bit wide average $\widetilde{P}_a$ of the absolute value of I/Q samples replaces the average $\widetilde{P}_s$ of power of I/Q samples, given

$$\widetilde{P}_a = \frac{1}{M}\sum_{M}(|I| + |Q|), \tag{9}$$

then $\widetilde{P}_s = A\widetilde{P}_a$

where $A = \frac{\sigma}{\sqrt{2\pi}}$

Proof of the equation above is based on the following assumption: the strength of CDMA I or Q signal has a zero mean normal distribution [3], namely $I, Q \sim N(0,\sigma)$. And the strength of I signal always equals to that of Q signal:

$$\frac{1}{M}\sum_{M}|I| = \frac{1}{M}\sum_{M}|Q|. \tag{10}$$

while $$S_a = \int_{-\infty}^{+\infty}|x|N(0,\sigma)dx = \sqrt{2\pi}\sigma \tag{11}$$

and $$S_s = \int_{-\infty}^{+\infty}x^2 N(0,\sigma)dx = \sigma^2 \tag{12}$$

B. Feed Backward

Fig.5 shows the practical structure of FBAGC in FPGA. The power estimation is done at the scaled output. The algorithm of power estimation is the same as that in FFAGC. The LUT is used here to convert the power error into adjusting factor for every input sample. The adjusting factors are then accumulated to get the final forward path gain. In logarithm domain, the accumulation is done through an integrator; in non-logarithm domain an accumulating multiplier calculates it.

Hence the final gain is the result of times of multiplying of each adjusting factor, and consequently the final relatively stable output is the result of times of adjustments.

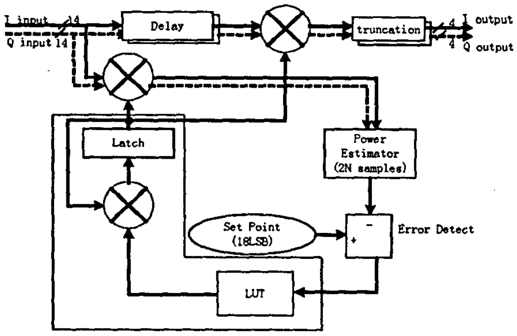

Figure 5. Practical structure of FBAGC

If the adjusting factor is a fixed value, it is inevitable for the feed back loop to introduce the AM (Amplitude Modulation) distortion after settling. To minimize the AM distortion and meanwhile maintain AGC response fast enough, the adjusting factor is set to be adaptable. Namely, adjusting factor is a cluster of factors proportioned to the errors, as the error decreases, the size of the adjustment decreases. This causes the loop gain to asymptotically approach the final gain value. For the same reason above, to prevent from vibration of the output due to excessive adjustments before settling, the largest adjusting factor should also be set small enough. Given $k_{\max}$ is the largest adjusting factor, the following inequity is necessary:

$$k_{\max} \le e/M \tag{13}$$

where e is the error, M is the number of averaging samples for power estimation.

III. Performance

Each of the two structures has its inherent features. As described previously, the basic difference between the two structures results in the diversities of their performance.

Firstly, FFAGC calculates the final adjusted result as soon as it obtains the discrepancy of power level between input and set point. In another word, it directly obtains the final gain from the input with just one step. Whereas FBAGC calculates from the output every corresponding adjusting factor, which, combining all its previous adjusting factors, approaches its final gain, that is, the final adjusting factor is a result of accumulation of many steps adjustments, the response time of FFAGC is thereby definitely shorter than that of FBAGC.

Secondly, an excessive AM distortion after settling may be introduced by the feedback loop in FBAGC, in this case, the loop settles to where half of the error samples are positive and half are negative (in logarithm domain). Adaptable adjusting factors proportioned to the errors can minimize the distortion.

Thirdly, a large LUT is necessary in FFAGC to meet the need of large ratio of dynamic range to adjusting resolution, while FBAGC needs a relatively small LUT to obtain the final loop gain.

Fourthly, time delay of both AGCs can effectively reduce the periods of overflow and underflow, which can be seen from the following illustration.

Fig.6A to 9B show result of MATLAB simulation of fixed-point module of both implemented AGCs. Tab.1 is a group of practical adjusting factors corresponding to their set errors (the difference between the expected power estimate and the set point).

Fig.6A and 6B show the functionality of module FFAGC and FBAGC respectively. The input is QPSK signal amplitude modulated with a sine wave of very slow frequency emulating the power level variance. The modulated input is quantified through a 14-bit ADC. Fig. 7A and 7B show the corresponding constellation maps of I/Q modulated signals before and after AGCs.

Fig.8A and 8B show the performance of FBAGC without and with time delay of 64 samples respectively. Here the averaging length of power estimation of FBAGC is 64 samples. The input is a sine wave with step-varying power level, the increment and the decrement are about 12dB. The output power level of path I maintains to 9 LSB after settling (half of power demand $I^2+Q^2=18$ LSB). Fig.9A and 9B show the performance of FFAGC without and with time delay of 64 samples respectively. The averaging length of power estimation here is 128 samples, the input are the same with that of Fig. 8. Compared to Fig.8A and 9A, both outputs in Fig.8B and 9B have shorter periods of overflow and underflow as well as suppressions on the surges of output power level when step variance in input power level occurs.

IV. Conclusion

Two practical hardware implementations are discussed and their performances are compared in detail through MATLAB fixed-point simulation, which, in turn directs the implementation. Both AGCs meet well the system requirements of CDMA base station receiver. The algorithms developed in the design are feasible and effective in area-efficient design in FPGA.

References

[1] Proposed Publication Version of the Base Station Minimum Performance Standard (TIA/EIA-97-D), Seattle, WA,USA, 30 March 2001

[2] Architecture for 800 MHz CDMA Receiver, IF to Baseband 80-D5040-1, Qualcomm.

[3] CDMA RF System Engineering, Samuel C. Yang, Artech House

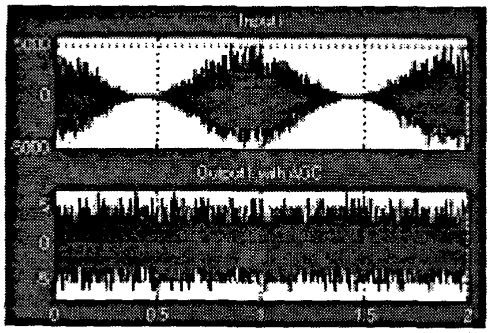

Figure 6A. I output with FFAGC

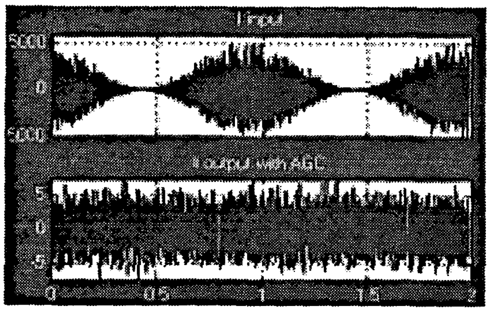

Figure 6B. I output with FBAGC

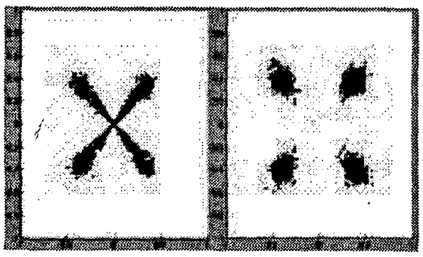

Figure 7A. I/Q display without (left) and with (right) FFAGC

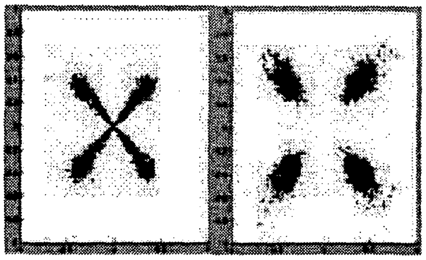

Figure 7A. I/Q display without (left) and with (right) FBAGC

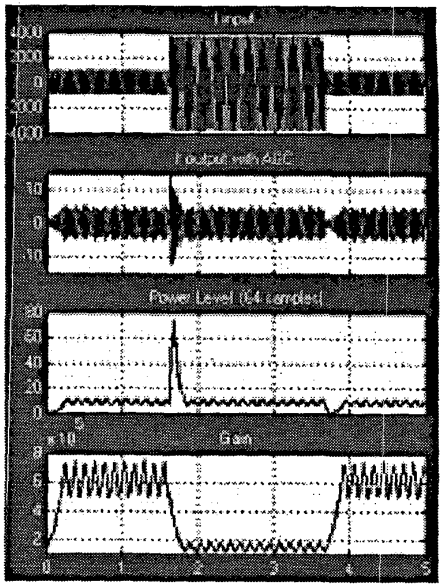

Figure 8A. FBAGC without delay

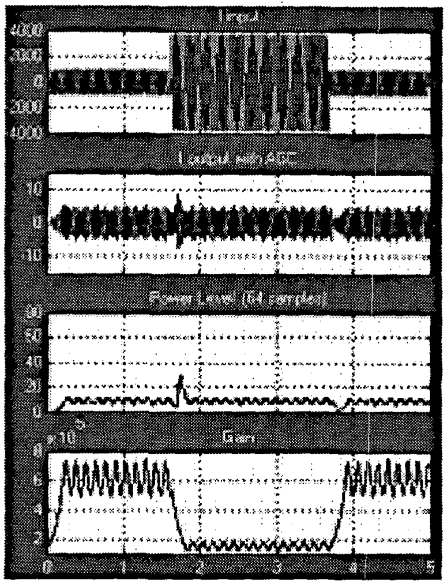

Figure 8B. FBAGC with delay of 64 samples

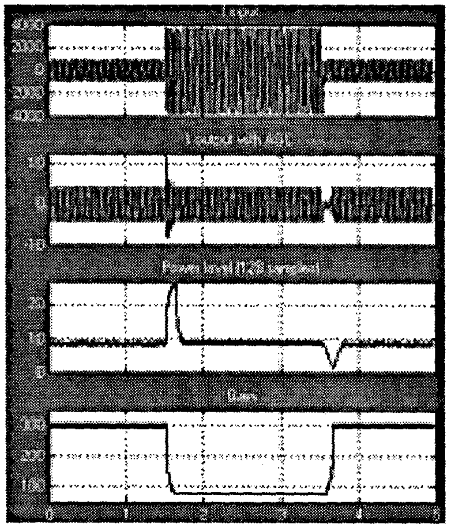

Figure 9A. FFAGC without delay

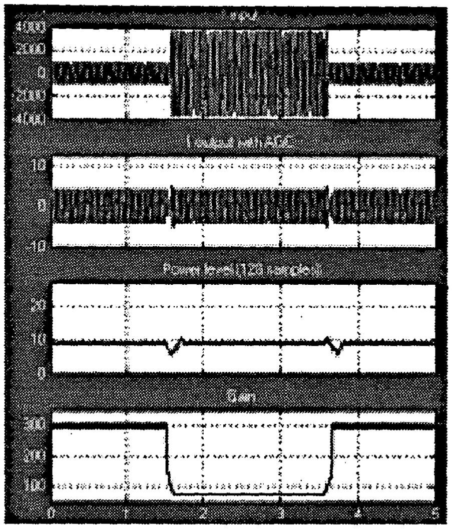

Figure 9B. FFAGC with delay of 64 samples

TABLE I. ADJUSTING FACTORS IN FBAGC

Error(dB)	Adjusting factor (dB)
0.1015625000	-0.0031738281
0.0338541667	-0.0010579427
0.0112847222	-0.0003526476
0.0037615741	-0.0001175492
0.0012538580	-0.0000391831
0.0004179527	-0.0000130610
0.0001393176	-0.0000043537
0.0000464392	-0.0000014512
0.0000000000	0.0000000000
-0.0000464392	0.0000014512
0.0001393176	0.0000043537
-0.0004179527	0.0000130610
-0.0012538580	0.0000391831
-0.0037615741	0.0001175492
-0.0112847222	0.0003526476
-0.0338541667	0.0010579427
-0.1015625000	0.0031738281