Design of Chip Set for CDMA Mobile Station

Kwangil Yeon, Hayoung Yoo, and Kyungsoo Kim

CONTENTS

ABSTRACT

In this paper, we present a design of modem and vocoder digital signal processor (DSP) chips for CDMA mobile station. The modem chip integrates CDMA reverse link modulator, CDMA forward link demodulator and Viterbi decoder. This chip contains 89,000 gates and 29 kbit RAMs, and the chip size is 10 mm×10.1 mm which is fabricated using a 0.8 μm 2 metal CMOS technology. To carry out the system-level simulation, models of the base station modulator, the fading channel, the automatic gain control loop, and the microcontroller were developed and interfaced with a gate-level description of the modem application specific integrated circuit (ASIC). The Modem chip is now successfully working in the real CDMA mobile station on its first fab-out. A new DSP architecture was designed to implement the Qualcomm code exited linear prediction (QCELP) vocoder algorithm in an efficient way. The 16 bit vocoder DSP chip has an architecture which supports direct and immediate addressing modes in one instruction cycle, combined with a RISC-type instruction set. This turns out to be effective for the implementation of vocoder algorithm in terms of performance and power consumption. The implementation of QCELP algorithm in our DSP requires only 28 million instruction per second (MIPS) of computation and 290 mW of power consumption. The DSP chip contains 32,000 gates, 32k (2k×16 bit) RAM, and 240k (10k×24 bit) ROM. The die size is 8.7 mm×8.3 mm and the chip is fabricated using 0.8 μm CMOS technology.

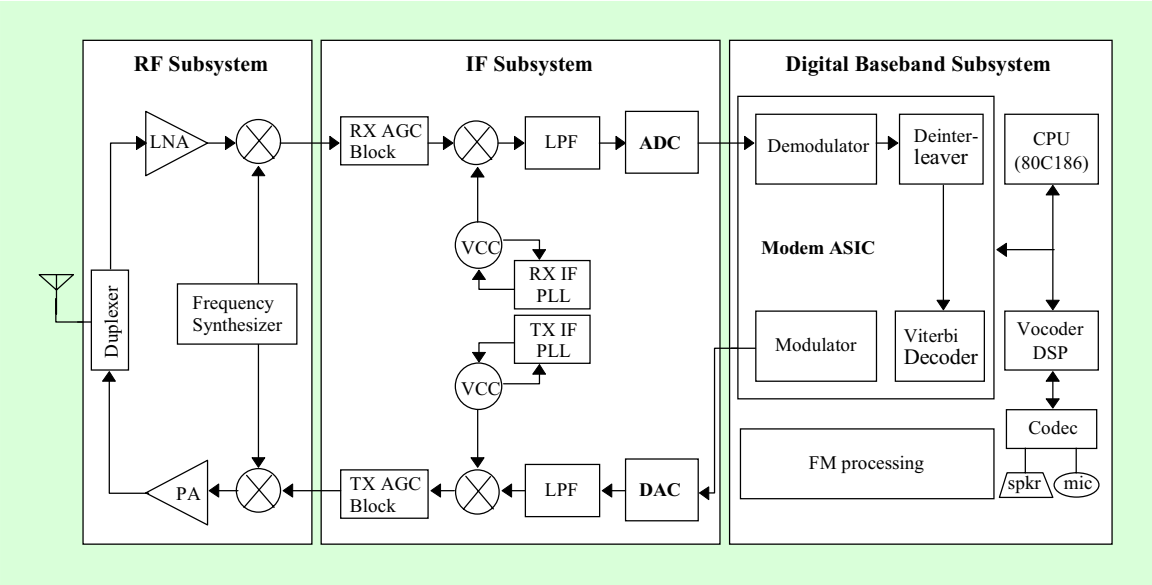

Fig. 1. Block diagram of CDMA cellular mobile station.

I. INTRODUCTION

Digital cellular communications are increasingly replacing analog counterparts because they offer improvement in channel capacity, voice quality, noise immunity, and security [1]. The CDMA scheme specified in TTA-62 [2] has been adopted as the standard of Korea digital cellular system. The CDMA digital cellular telephone shown in Fig. 1 is composed of radio frequency (RF) subsystem, intermediate frequency (IF) subsystem, and digital baseband subsystem. Key hardware components of the digital baseband subsystem include modem ASIC, 16-bit DSP which performs Qualcomm code exited linear prediction (QCELP) variable rate vocoding algorithm, and 16-bit microcontroller. The FM processing unit is also included in the digital subsystem and IF subsystem for CDMA/FM dual-mode telephone.

We designed a CDMA modem ASIC which integrates reverse link modulator, forward link demodulator, and Viterbi decoder required for the CDMA baseband digital signal processing. It is controlled by the microcontroller for timing and data processing. We also designed a DSP chip to implement the QCELP vocoder algorithm to meet the requirement of low power consumption. The chip set was fabricated using 0.8 μm CMOS technology and fully tested in a real CDMA mobile station.

In this paper, we present the design and the implementation of the CDMA modem ASIC and the vocoder DSP chip. We also describe the design methodology and

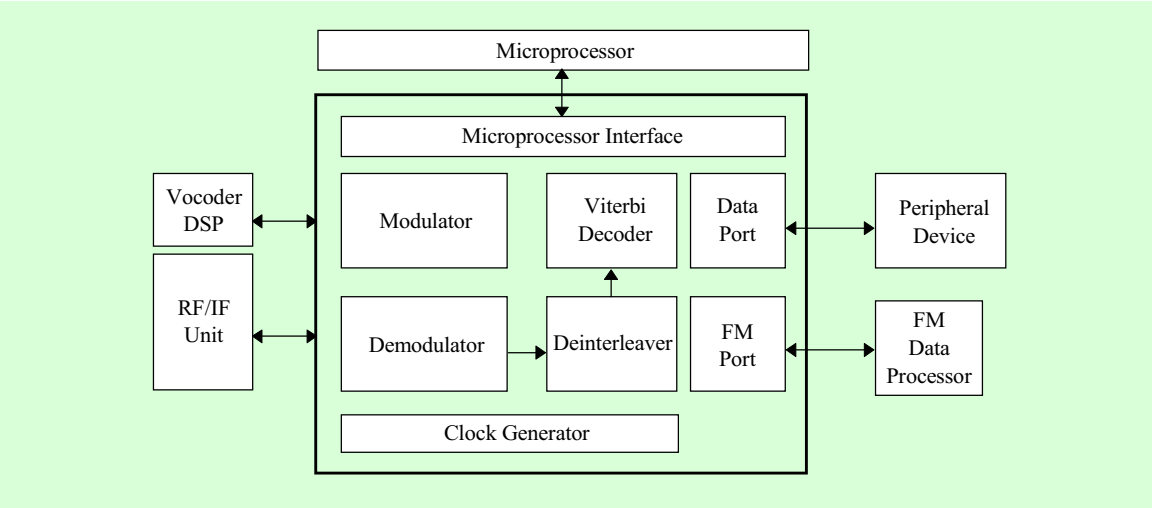

Fig. 2. Block diagram of CDMA modem ASIC.

the CAD environment used for pre-silicon system-level verification of the chip set.

II. DESIGN OF THE MODEM ASIC

The modem ASIC performs reverse-link modulation and forward-link demodulation of spread spectrum signal [3]. It consists of the following functional modules: modulator, demodulator, deinterleaver, Viterbi decoder, and some pheriperal blocks. Figure 2 shows the overall functional block diagram of the modem ASIC.

1. Modulator

The reverse channel is composed of access channels and reverse traffic channels. The data transmitted on the reverse channel are grouped into 20 ms frames. The modulation includes convolutional encoding ($R=1/3$, $K=9$), block interleaving, 64-ary orthogonal modulation, 42-bit long pseudo noise (PN) code spreading, offset quadrature phase shift keying (OQPSK) spreading modulation, and FIR filtering. The modulator block is structured as shown in Fig. 3.

After adding cyclic redundancy code (CRC) for both 9,600 bps and 4,800 bps rates and adding eight encoder tail bits, the 20 msec data packet is transmitted on the reverse traffic channel at the data rate of 9,600, 4,800, 2,400, or 1,200 bps. The transmitted data are encoded using a rate 1/3 convolutional code with constraint length of 9. This process generates 3 coded symbols for each data bit. The encoder outputs are repeated before interleaving when the data rate is lower than 9,600 bps. The repeated symbols shall be deleted prior to actual transmission by data burst randomizer.

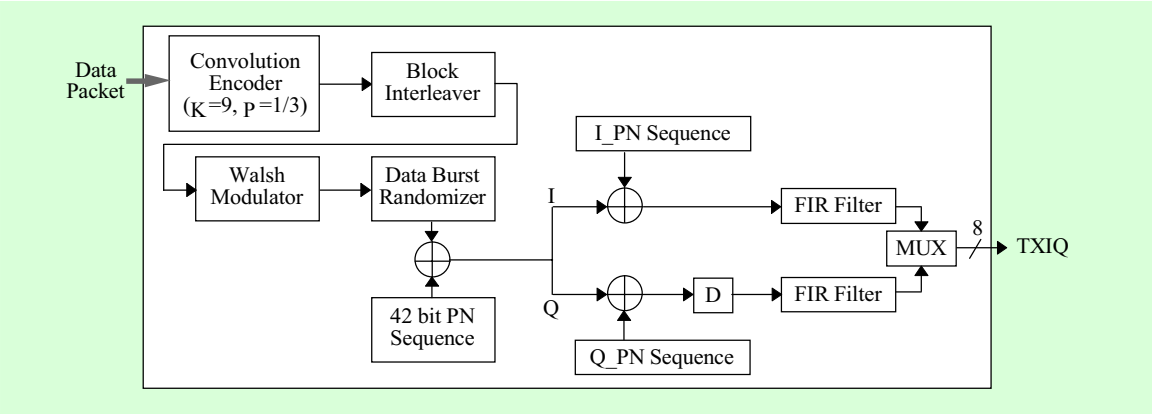

Fig. 3. Functional block diagram of modulator.

The encoded symbols are then interleaved over a 20 ms interval. The interleaver is a memory array with 32 rows and 18 columns. The coded symbols are written into the interleaver by columns and read out by rows in a pre-determined order. The interleaved data are grouped to six symbol groups and then converted into one of 64 Walsh orthogonal modulation symbols.

The data burst randomizer generates a masking pattern to mask out the redundant data generated by the symbol repetition. The masking pattern is determined by the data rate and by a block of 14 bits in the long code. The Walsh chips are then spread by the long 42-bit PN codes produced by a PN sequence generator. When transmitting on the traffic channel, the mobile station uses one of two long code masks unique to that mobile station. After long PN spreading, there is another short PN spreading which is quadrature spreading using In (I)-phase and Quadrature (Q)-phase short PN codes produced by two 15-bit PN generators. The data spread by the Q-phase PN sequence is delayed by half a PN chip time with respect to the data spread by the I-phase PN sequence for OQPSK modulation.

Following the spreading operation, the I and Q data are input to the baseband FIR filters. These 48-tap FIR filters have symmetrical impulse response and linear phase response. The passband ripple is less than 1.5 dB and the stopband attenuation is 40 dB. The output of this filter is oversampled four times and interpolated.

2. Demodulator

The forward channel consists of pilot channel, sync channel, paging channels, and forward traffic channels. Each of these code channels is orthogonally spread by appropriate Walsh function and is then spread by a quadrature pair of PN sequences. Demodulation process performs complementary operations to the above modulation

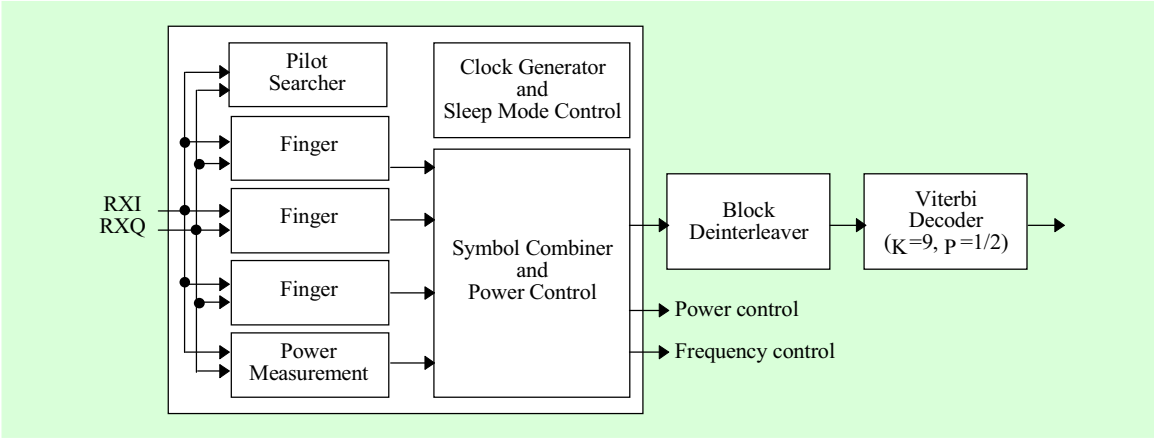

Fig. 4. Functional block diagram of demodulator.

process on the forward channel. It includes synchronization of the receiver carrier's phase and frequency as well as its chip timing to those of the transmitted signal. Figure 4 shows a functional block diagram of demodulator which includes searcher, fingers and symbol combiner to perform synchronization and demodulation of multipath received signals.

The searcher block performs the acquisition process by scanning and estimating the signal strength at each pilot PN sequence offset. It involves testing all likely hypotheses by correlating a locally generated PN signal with the received PN signal. Double dwell serial search strategy is used to minimize the searching time. Each hypothesis is first checked for a fixed time L1 against a predefined threshold T1. If it fails, it proceeds to check the next hypothesis. If it succeeds, a larger dwell L2 is checked against threshold T2. Each of these energies is sent to the microcontroller over a DMA channel and sorted in the values of energies. The microcontroller uses this information to select a pilot signal.

A three-finger rake receiver is used for demodulation of multipath fading signals. Each of the fingers performs short PN despreading, orthogonal Walsh demodulation, pilot energy measurement, timing deskew, frequency error signal generation, and timing tracking. Each of these fingers can track a single pilot by itself and maintains timing independent of the other fingers. The microcontroller keeps track of the finger time and energy values and assigns new pilot offsets to that finger as needed to track the pilots with the most energy as measured by the searcher.

The symbol combiner block takes the role of combining finger outputs, extraction of power control bit, long PN despreading, and reference clock generation. It also keeps track of the absolute system time and tries to keep the fingers within 4 symbols of each

other. This block also generates power control and frequency control signals which are sent to the IF analog circuitry as pulse density modulated signals. The demodulation process extensively interacts with the microcontroller, and its performance is closely related to the microcontroller software.

3. Deinterleaver and Viterbi Decoder

The output stream from the demodulator is de-interleaved using the deinterleaver block which is implemented using RAM module inside the modulator. The de-interleaved stream is passed to the Viterbi decoder which is used to optimally decode synchronized and quantized symbol stream. The Viterbi decoder is composed of input buffer block, branch metric block, add-compare-select (ACS) block, traceback module, and output buffer block [4]. The input buffer stores soft decision symbol data received from the deinterleaver. The symbols are processed by ACS block, and the results are stored as state metrics in internal RAM module. Decisions obtained from the ACS operation are saved into a path memory module, and a traceback process through this path memory outputs a single data bit. This decoder produces 4.5 dB coding gain for the rate 1/2 of constraint length $K = 9$ convolutional code. It also calculates a symbol error rate and checks the frame CRC code to produce a quality bit. The decoded data bits with quality information are then transferred to the microcontroller.

4. Design and Verification Methodology

The top-down ASIC design begins with an ASIC requirement specification, followed by the behavioral verification of system/ASIC algorithms. However, analyzing the behavior of ASIC may not be sufficient to detect all the errors in the circuit. In such cases, register transfer level (RTL) or gate-level description is more appropriate for pin-pointing the errors related to the critical behavior of the ASIC. Considering all the above possibilities makes mixed-level simulation with VHDL simulator as the best candidate for the overall system simulation.

The system-level verification method focuses on modeling of the target system rather than ASIC components [5]. We tried to emulate the operation of the target system by running its application software on the simulation models, thereby checking its actual function in a virtually real situation. Even though this method has some limitations, this kind of logic emulation may be a key technology for verifying the system and ASIC performance before implementation and for debugging the system hardware/software. For this purpose, we developed VHDL models of Intel 80C186 CPU [6] and a memory unit, and rake receiver input patterns generated using signal processing workbench (SPW) tool [7]. Overall verification environment for CDMA mobile system shown in Fig. 5 consists of base station modulator C model, Rayleigh fading

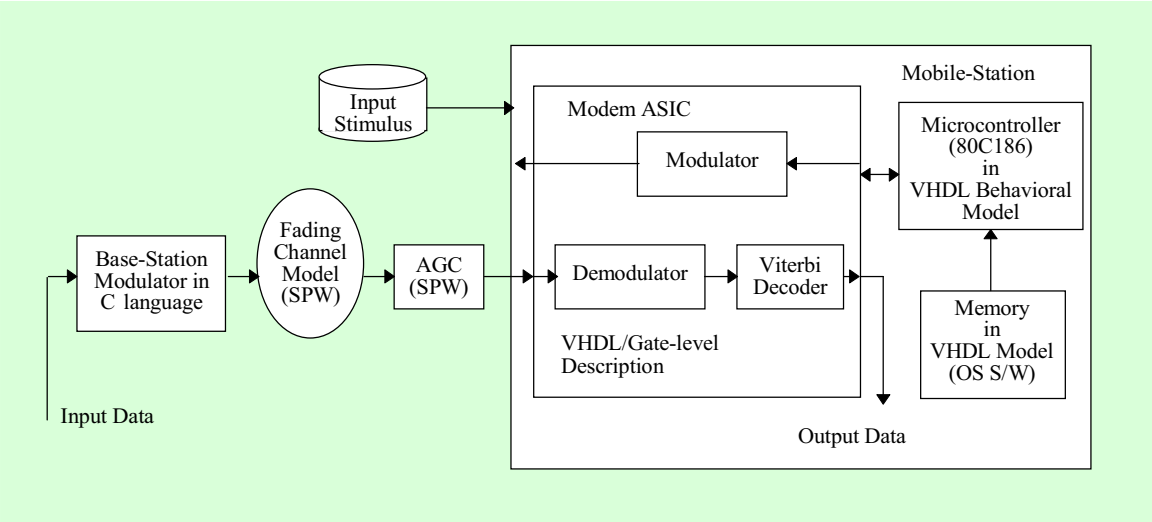

Fig. 5. Verification environment for CDMA mobile station.

channel & AGC loop model [8], 80C186 microcontroller VHDL model, and a simplified version of operation software.

With the above modeling, system-level simulation of the modem ASIC is performed by executing the actual application software. We could identify a number of errors in the RTL description of the functional modules while performing system simulation. Most of these mistakes stem from the lack of understanding on the proper interfaces between the microcontroller and the ASIC as well as among the functional blocks themselves. After all the necessary corrections, the system was successfully initialized and commenced its proper operation.

When we used Compass software simulator [9] to simulate the mobile station for 40 ms of real time, it took about 30 hours on SPARC20 workstation. By using the IKOS NSIM64 hardware accelerator [10] connected to the VHDL simulator, we could simulate 7 superframes (560 ms in real time) within 27 hours which is minimally necessary to confirm the call processing of the CDMA system.

5. ASIC Implementation and Test Results

At the initial stage of this project, we developed C models for major functional blocks of the CDMA modem and performed behavioral-level simulations. Using these high-level simulation results, we understood the major functions of this chip and decided the architecture of the modem chip. At the next stage, we designed each block using gate-level or RTL-level description. Our system-level verification environment was extensively used to verify the design. We used a 0.8 μm 2 metal CMOS standard-cell library and compiled cells for

RAM and datapath blocks. The total number of gates was about 89,000 gates and 29k RAMs. The chip size was 10 mm×10.1 mm and it was packaged in 144-pin thin quad flat package (TQFP). The power consumption is about 340 mW in CDMA transmitting/receiving mode at 10 MHz and 33 mW in sleep mode. The fabricated chip was fully tested and successfully worked in the real CDMA mobile station on its first fab-out. Figure 6 shows a micro-photograph of the CDMA modem chip.

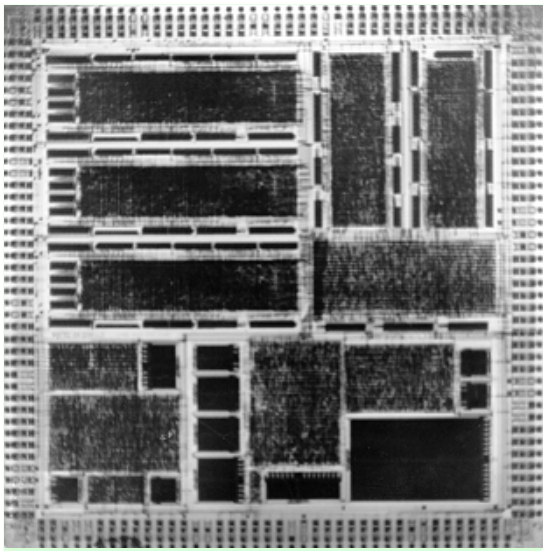

Fig. 6. Micro-photograph of the CDMA modem chip.

III. DESIGN OF THE VOCODER DSP

1. Overview of QCELP Algorithm

The QCELP algorithm defined in IS-96 [11] has been selected as the vocoder algorithm for the CDMA system. A different feature of the QCELP algorithm from general CELP algorithm is that it changes the compression rate from 8 kbps to 1 kbps depending on the voice activity. Figure 7 shows a block diagram of QCELP speech coder.

In the encoding process, linear predictive coding (LPC) coefficients are calculated from formant characteristics of the voiced sound. The LPC parameters are transformed into line spectral pair (LSP) frequencies and these frequencies are converted into LSP codes for transmission. The rate decision process selects the data rate based on the energy in the frames. And the pitch parameters (pitch gain and lag) are determined by using an analysis-by-synthesis method. The codebook parameters are also searched and selected to minimize the residuals of LPC and pitch search process. Finally, the LSP codes, pitch parameters, and codebook parameters are packed into packet data and transmitted to the CPU. The decoding process synthesizes μ-law speech signal from the encoded packet using a reverse operation of the encoding process.

To achieve the low power consumption for mobile applications, efficient implementation method of the algorithm is as important as the DSP design. We designed a new DSP chip and an improved method of implementing the pitch searching process which occupies almost 50 % of the total computation time [12].

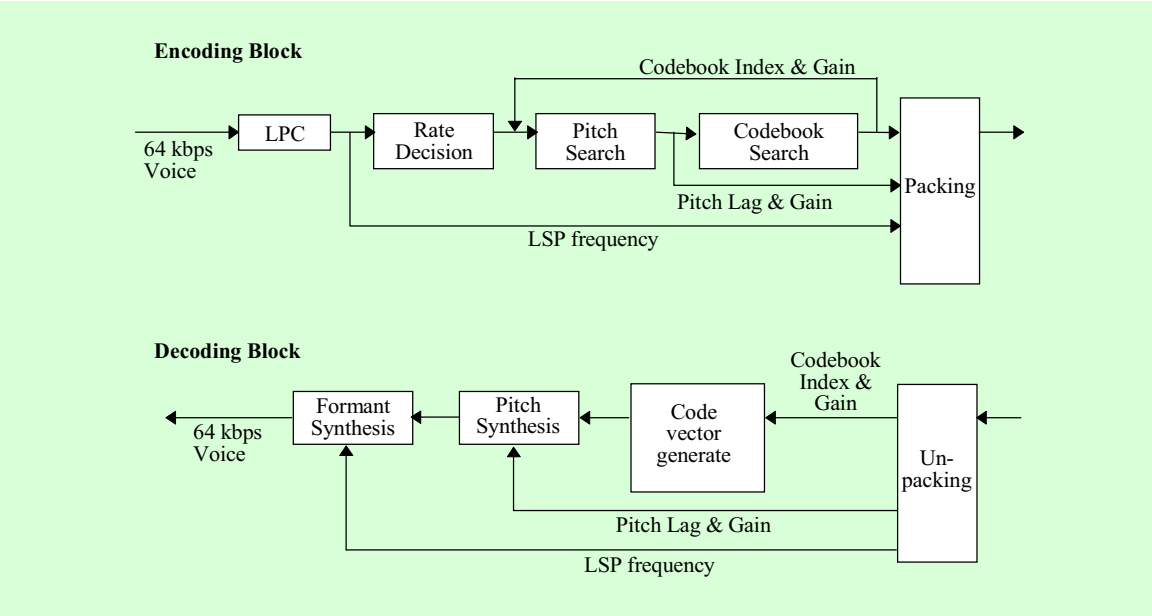

Fig. 7. QCELP speech coder.

2. Overview of DSP Architecture

To implement the QCELP vocoder algorithm for the mobile application, DSP must have fast operating capability more than 30 MIPS and low power consumption. To meet the requirements mentioned above, We adopted a RISC type instruction set, distributed decoding, alternative program fetch, repeat control and dual bank memory system in our DSP architecture [13]-[15]. Moreover, by adopting 24 bit instruction format, the immediate and direct addressing modes can be coded in one instruction word. This feature enables execution of all instructions in one clock cycle.

The architecture of our DSP shown in Fig. 8 consists of four major functional blocks: program control block, memory block, ALU block, and I/O block.

The program control block includes program ROM, instruction register, and program stack. The memory block includes data RAM, ROM, address registers, stack pointer register, and index registers. The ALU block contains ALU, multiplier, and barrel shifter. And I/O block is composed of interrupt control registers, serial port registers, and parallel port registers.

A. Program Control Block

The main function of this block is sequencing the instruction flow using the program counter (PC) and instruction register (IR). The PC holds the program address of

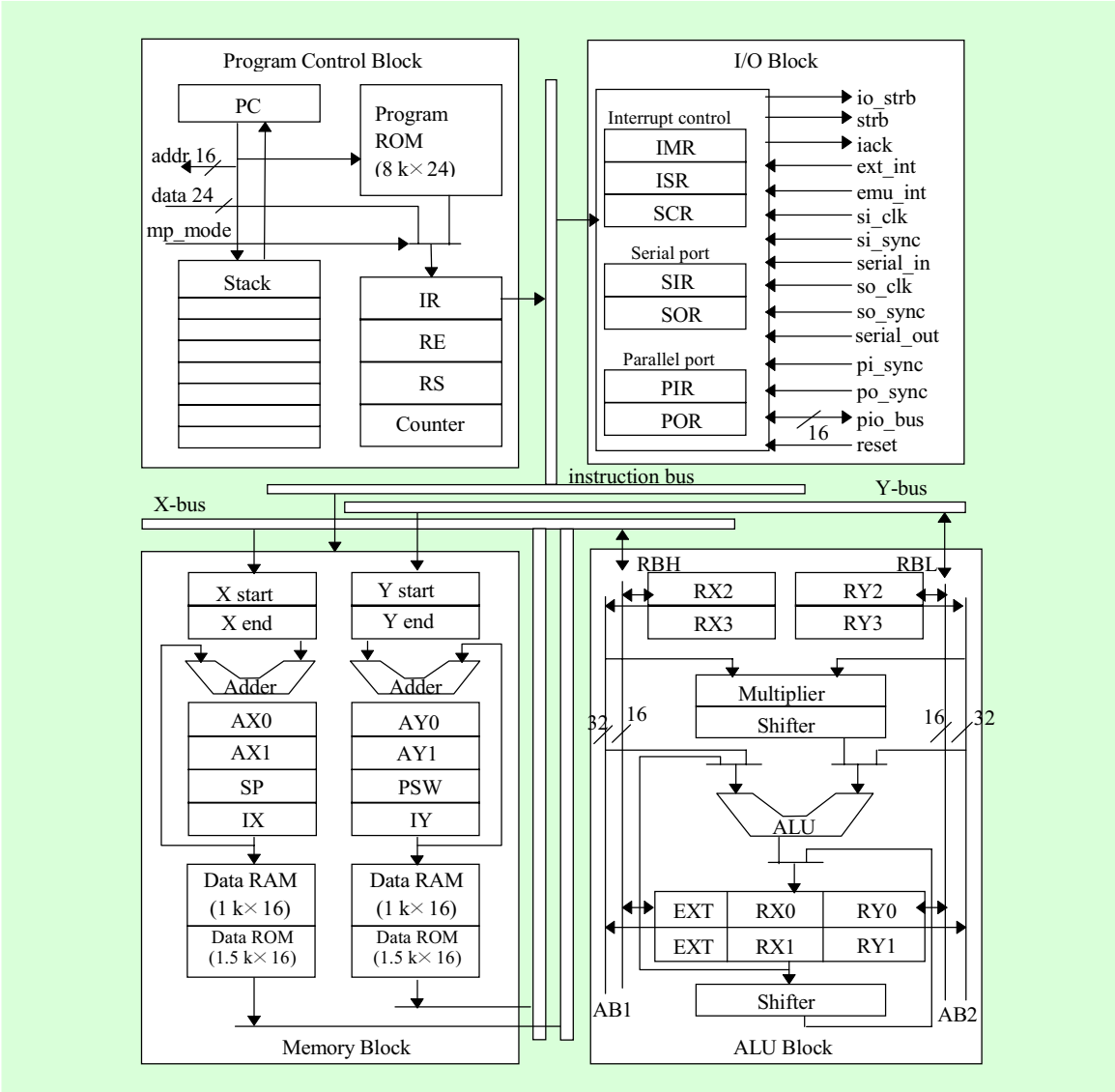

Fig. 8. Functional block diagram of DSP chip.

the instruction to be fetched from the program memory and the IR has the instruction word being decoded. The bit lengths of PC and IR are 16 and 24, respectively. The content of IR is broadcast to the other functional blocks for the separate decoding and execution depending on the instruction type. Besides PC and IR, the 8 k×24 bit program ROM, repeat start (RS) register, repeat end (RE) register, repeat counter (RC) and their 3-stage stacks for the nested repeat instruction are the main resources of

the block. The DSP has 3 pipeline stages for most of the instructions except multiply instructions which are executed in 4 pipeline stages.

B. Memory Block

This block is responsible for the data load /store operations between data RAM/ROM and the general purpose registers. The DSP supports immediate, register, direct and indirect addressing modes. Among them, the immediate and the register addressing modes are processed in ALU block. And the direct and the indirect addressing modes which require memory access are processed in this block. The DSP has the dual bank (x-bank, y-bank) data memory structure which can support two 16-bit word operands load or store operations at the same time. In memory access operation, address registers ($ax0$, $ax1$) in the x-bank address XRAM or XROM, and address registers ($ay0$, $ay1$) in the y-bank address YRAM or YROM. The $x(y)$ bank RAM/ROM and the $y(x)$ bank registers can communicate without any overhead except for the dual data load/store operation where the $x(y)$ bank RAM/ROM can only access the $x(y)$ registers. Each bank has 1 k×16 bit RAM, 1.5 k×16 bit ROM, two address registers, and an index register to support the register indexed indirect addressing mode operation. Stack pointer (SP) and stack pointer buffer (SPB) belong to this block to support the stack operation such as push and pop.

C. ALU Block

ALU block executes arithmetic/logic instructions such as add, subtract, multiply, AND, OR, and Exclusive OR, etc. This block consists of two 36 bit accumulators, four general purpose 16 bit registers for separate x/y bank, 18×18 multiplier and 36 bit barrel shifter. The accumulators can be divided into 16 bit X register, 16 bit Y register, and 4 bit extension nibble which covers arithmetic overflow range. The accumulators can also be used as the general purpose registers. The ALU block supports double precision ALU operations, the parallel ALU operations, and data move operation. In the parallel instruction, the ALU operation and data move operation are executed at the same time. In order to accomplish this operation, the ALU block has two separate 16 bit buses (RBL, RBH) for data move. X-bus in top block communicates with RBH bus and Y-bus communicates with RBL bus. Two separate 32 bit arithmetic buses (AB1, AB2) are used for single or double ALU instructions. In the single ALU instructions, operand of X-register (RX0-RX3) uses upper part of AB1 and operand of Y-register (RY0-RY3) uses lower part of AB2. In the double instructions, first 32 bit operand uses AB1 bus and second 32 bit operand uses AB2 bus.

D. I/O Block

This block consists of clock generators and interrupt control unit. The clock generator supplies the two groups of synchronized clock signals. One group goes to the

interrupt control block, and the other group is used for main functional blocks. Idle instruction disables the main clock generator to reduce the power dissipation. To awake the DSP, the user should request interrupt which is not masked. There are seven types of interrupts; reset, serial input, serial output, parallel input, parallel output, external, and emulation. The reset and emulation interrupts are non-maskable interrupts. The emulation interrupt is designed for software emulation purpose. The parallel port and the serial port have 8/16 bit communication modes.

3. Software Development Environment

Assembler was written in C language to translate the instruction-level assembly language into machine code. We also developed a software simulator to verify the instructions of DSP and to debug assembly language programs. This program provides the environment to develop an application software or firmware. We are now developing a C compiler to support the user with high-level programming environment.

4. ASIC Implementation

After deciding the architecture of this chip to implement the vocoder algorithm in an efficient way, we designed each block using gate-level or RTL-level description. We used 0.8 μm CMOS standard-cell library and compiled cells for RAM, ROM, multiplier, and datapath blocks. The total number of gates was about 32,000 gates, 8 k×24 bit program ROM, 3 k×16 bit data ROM, and 2 k×16 bit data RAMs. The size of chip shown in Fig. 9 was 8.7 mm×8.3 mm and the chip was packaged in 100-pin TQFP.

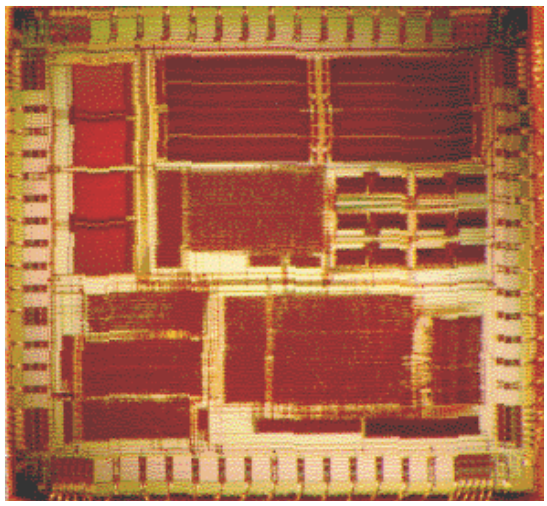

Fig. 9. Micro-photograph of DSP chip.

The maximum operating performance is about 40 MIPS at 80 MHz. We implemented the QCELP algorithm in our DSP chip and it required only 28 MIPS of computation and 290 mW of power consumption. The fabricated chip was fully tested and successfully working in the real CDMA mobile station.

IV. CONCLUSIONS

We presented the design of CDMA modem ASIC with design methodology and CAD environment which is used for system-level verification of ASIC in VHDL system model. The modem ASIC which consists of

89,000 gates and 29 k RAMs is now successfully working in the real CDMA Mobile Station on its first fab-out. To carry out the system-level simulation, the high-level models of the base station modulator, the fading channel, the AGC loop, and the microcontroller were developed and interfaced with the RTL description of the modem ASIC.

We also presented a new DSP chip which was designed to implement the vocoder algorithm in an efficient way. The implementation of the QCELP algorithm in our DSP chip requires only 28 MIPS of computation and 290 mW of power consumption. Since the power consumption is the most important issue for the mobile application, the development of more efficient algorithm and low power architecture will be the future research direction.

ACKNOWLEDGMENTS

The authors would like to thank CDMA ASIC & DSP design team members for their contribution to the design of the chip set, and to Jaesoek Kim, Hoonbock Lee, Hyungmoo Park, and Sinjong Park for their helpful comments and support for this work.

REFERENCES

[1] K. S. Gilhousen *et al.*, "On the capacity of a cellular CDMA system," *IEEE Trans. on Vehic. Tech.,* vol. 40, no. 2, pp. 303-312, 1991.

[2] TTA-62 : *Interim standard for common air interface for digital cellular in the 800MHz band*, 1994.

[3] K. Yeon, I. Eo, I. Jeon, I. Lim, J. Kim, and H. Lee, "Design of modem ASIC for CDMA mobile station system," *Proc. KITE Fall Conf. '94,* vol. 17, no. 2, pp. 1431-1434, 1994.

[4] I. Jeon, I. Eo, I. Lim, K. Yeon, J. Kim, and H. Lee, "Viterbi decoder performance analysis in CDMA digital cellular system," *Proc. KICS Fall Conf.,* vol. 13, no. 2, pp. 431-434, 1994.

[5] G. Park, K. Chang, J. Kim, and K. Kim, "System-level verification of CDMA modem ASIC," *Proc. ASP-DAC95 & VLSI95*, Japan, Aug. 29 - Sept. 1, 1995, pp. 177-182.

[6] B. B. Brey, *The Intel Microprocessors*. MacMillan Publishing Company, 1991.

[7] Alta Group of Cadence, *SPW User's Manual,* March 1995.

[8] K. Chang, I. Jeon, I. Eo, X. Lin, and J. Kim, "Frequency selective Rayleigh fading channel & AGC loop modeling for CDMA," *Proc. APCC '95,* Japan, June 13 -16, 1995, pp. 240-244.

[9] Compass Design Automation, *Mixed-mode Simulator,* 1992.

[10] IKOS Systems, *Voyager Series User's Guide,* Oct. 1994.

[11] TIA/EIA IS-96, *Speech service option standard for wideband spread spectrum digital cellular system,* TIA, Apr. 1994.

[12] K. J. Byun *et al.*, "On a reduction of pitch searching time by separating the speech components in the CELP vocoder," *Journal of Acoustical Soc. of Korea*, vol. 14, no. 1E, pp. 22-29, 1995.

[13] K. C. Han. J. J. Kim, H. Y. Yoo, K. J. Byun, D. K. Kim, M. J. Bae, J. S. Kim, and H. B. Lee, "Design of DSP for vocoder application," *Proc. KITE Fall Conf. '94,* vol . 17, no. 2, pp. 1494-1497, 1994.

[14] Emmanuel C. Ifeachor and Barrie W. Jervis, *Digital Signal Processing a Practical Approach.* Wokingham: Addison-Wesley, 1993.

[15] Thomas W. Parsons, *Voice and Speech Processing.* New York: McGraw-Hill Publications Co., 1976.

Kwangil Yeon received the B.S and M.S degrees in electronics engineering from Hanyang University, Seoul, Korea in 1981 and 1984, respectively. He joined ETRI in 1982 and worked mainly on circuits and VLSI design . He is currently the Head of Telecommunication Circuits Section. His research interests include wireless modem design for CDMA cellular, PCS and wireless local loop.

Hayoung Yoo received the B.S degree in electronics engineering from Seoul National University, Korea in 1982. He obtained M.S and Ph.D. degrees in EE&CE from Iowa State University, U.S.A, in 1985 and 1988, respectively. He joined ETRI in 1989 and worked mainly on ASIC cell modeling and ASIC design. His research interests are VLSI architecture design and source coding for voice and multimedia.

Kyungsoo Kim received the B.S degree in electronics engineering from Sogang University, Seoul, Korea in 1977. He joined ETRI in 1977. He worked mainly on chip testing, process development and VLSI design. He is currently the Director of Integrated Circuits Department in Semiconductor Division of ETRI. His research interests are CAD software and ASIC design for wireless communication and ATM system.