Co-Emulation Design for OFDM Baseband Transceiver

Yang-Han Lee[1]*, Yih-Guang Jan[1], Ming-Hsueh Chuang[1], Hsien-Wei Tseng[1], Liang-Lin Jau[2], Min-Ru Wen[2], Wei-Chen Li[2] and Sheng-Kai Yu[1]

[1]*Department of Electrical Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.*
[2]*Department of Computer & Communication Engineering, St. John's University, Tamsui, Taiwan 251, R.O.C.*

Abstract

In this paper we use Top-down method to design WLAN802.11a based Orthogonal Frequency Division Multiplexing (OFDM) Baseband Transceivers. In this paper we emphasize the development of system hardware realization technique as the main issue and exploit Verilog hardware description language to complete the design of Convolutional encoder/Viterbi decoder, Mapper/Demapper and FFT/IFFT which all meet the timing pulse specifications of WLAN802.11a. In this paper we use Trace back architecture in Viterbi decoder design and 64 points of FFT/IFFT algorithm with radix 2^2 SDF architecture and control the connecting timing pulses between Mapper/Demapper and FFT/IFFT within 4μs of the same required sampling instant for the Mapper and Guard Interval to complete the desired transmission. In addition we will examine the effect due to finite bit length to determine the required bit length to have the minimum truncation errors. We emphasize in the hardware design. We utilize the Pathfinder hardware developed by Galaxy Far East Corp Company as our simulation platform and use Co-emulation method to verify and realize WLAN Orthogonal Frequency Division Baseband Transceiver system.

Key Words: Orthogonal Frequency Division Multiplex (OFDM), Fast Fourier Transform (FFT), Viterbi, Co-emulation, Pathfinder

1. Introduction

Due to the population of wireless area networks and users take the advantages of multimedia and other large volume information transmission the transmissions speed of IEEE 802.11b is unable to meet user's requirement. Instead it has been replaced by Orthogonal Frequency Division Multiplex [1] technique with IEEE 802.11a [2] and IEEE 802.11g [3] implemented systems to increase information transmission speed from 11 Mbps to 54 Mbps. In computer communication, the use of Wireless Local Area Network (WLAN) has the merits of reducing wiring connection problem, short installation time, high mobility, and wide coverage area etc. In recent years Asymmetric Digital Subscriber Line (ADSL), Very high speed Digital Subscriber Line (VDSL), Digital Audio Broadcasting (DAB) system and the European-standard based Digital Video Broadcasting (DVB) are all using OFDM technique. In addition in wireless communication it will employ the excellent OFDM characteristic of resisting multipath effect to solve the problem generated by the wireless communication channels. Because of this resistance to multipath effect in OFDM system, it draws all attentions of adopting OFDM system as the major communication system in the upcoming fourth generation of mobile communications. The theoretical development of wireless area network has somewhat matured but it is still scarcity in the actual system realizations. We try to

*Corresponding author. E-mail:691350101@s91.tku.edu.tw

design the system from the overall system point of view to integrate digital circuits to complete the design of base-band transceiver circuit.

2. Circuit Integrated Design for OFDM System

2.1 Finite Bit Length Effect

In hardware implementation it is important to consider the finite bit length effect in algorithm operations. Although it will reduce hardware cost by using short bit length implementation, its resolution will be sacrificed and consequently the system will endure higher quantization noise. On the other hand it will have better resolution by utilizing longer bit length. But in this situation it will increase the hardware complexity and also waste more energy and increase its execution time. It is important to find the best bit length.

Since FFT is a recursive operation the error accumulated from FFT continuous operations may be enormous it is necessary to find a method to determine the best bit length to be used in the FFT operation. First we need have a valid comparison standard to decide the FFT performance when it is implemented by using finite bit length. We use the resulting quantization error in FFT operation as our comparison standard. In Figure 1 it shows the resulting quantization errors in FFT when different bit lengths are used.

It uses SQNR (Signal to Quantization Noise Ratio) to compare the quantization noise generated between the ideal FFT and other fixed finite bit length FFT. The FFT SQNR has the following definition:

$$SQNR = 10\log\frac{\sum_{n=0}^{n-1} x_q(n)^2}{\sum_{n=0}^{n-1}(x_q(k) - \hat{x_q}(k)^2)} \quad (1)$$

By considering the combined factor of hardware cost and the overflow effect due to finite bit length we select the performance at SQNR = 45 dB [4] as our design goal for the implementation of FFT/IFFT. It concludes from Figure 2 that we need at least of 11 bits for the input data and 12 bits of bit length for each twiddle.

After we know the required bit length to be used in FFT we then try to find the best bit length to be used in Mapper/Demapper because it shows the existence of quantization error effect due to the usage of finite bit length in this block. In Figure 3 we present the format used to define any numerical number, the first bit is the sign bit, the second bit is the integer bit and its mantissa value is expressed by 9 bits. In Table 1, we compare the relation between the ideal value with its truncation error in representing 64-QAM.

From Table 1, it shows that the truncation errors are the same when it uses 11 or 12 bits in the representation, and as a matter of fact they all show the same errors values when we use 11, 12 or even up to 19 bits. In other words we will get the same accuracy when we use any bit length in this range interval. In order to reduce hardware cost, we select 11 bits as the bit length in the hardware implementation and we also use 11 bits to design BPSK, QPSK and 16-QAM signals. In order to demonstrate the system design we select 11 bits as the minimum bit

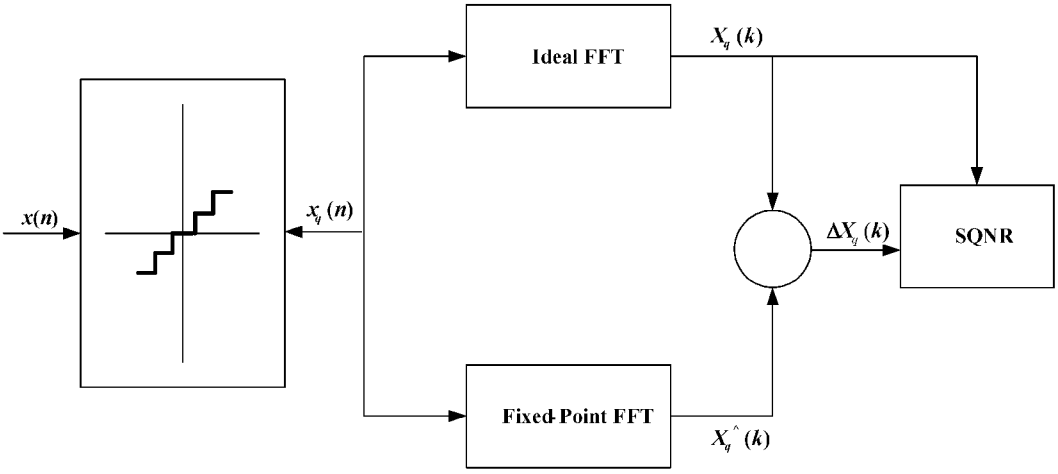

Figure 1. FFT quantization noise by using finite bit length.

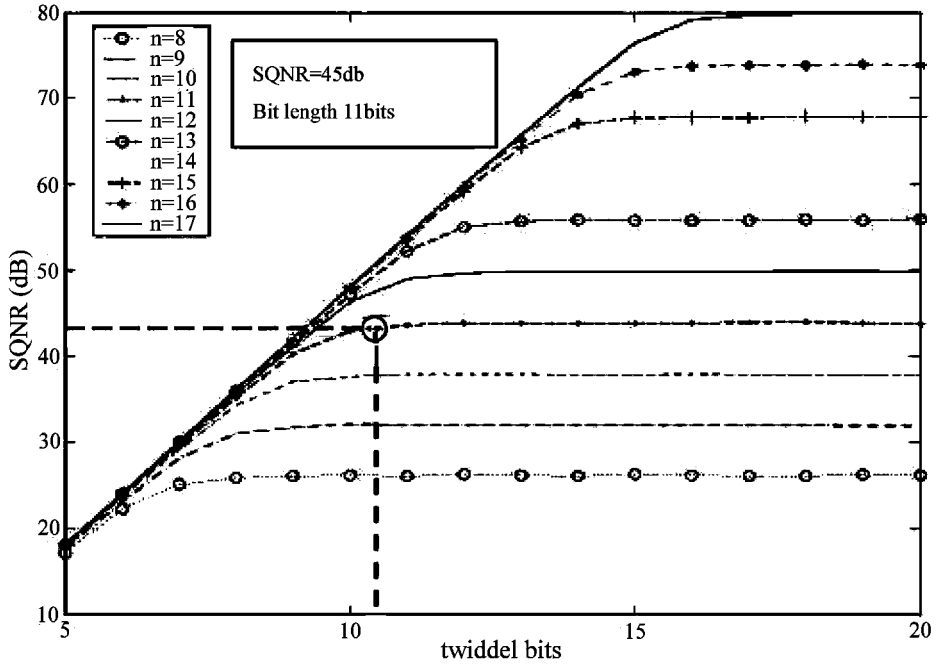

Figure 2. Simulation of SQNR vs. bit length.

Figure 3. Finite bit length representation of a numerical number.

Table 1. Comparison between ideal value and truncation error

		Normalization factor	Floating	Fixed (10b)	Fixed (11b)	Fixed (12b)
64QAM	1	$1/\sqrt{42}$	0.15430334996209	0.15234375	0.154296875	0.154296875
	3	$1/\sqrt{42}$	0.46291004988628	0.4609375	0.462890625	0.462890625
	5	$1/\sqrt{42}$	0.77151674981046	0.76953125	0.771484375	0.771484375
	7	$1/\sqrt{42}$	1.08012344973464	1.078125	1.080078125	1.080078125

length in the input of FFT/IFFT.

2.2 The Design of Timing Pulse in the Core System Transmission

Based on 802.11a communication functional blocks we use Verilog hardware description language to complete the implementation of important baseband circuits. We do not include the intercommunication between MAC and basband circuit. We do not consider the interconnection between analog circuits and the baseband transceiver either. We implement and test the designed baseband transceiver via FPGA.

We design and implement the following constituent functional blocks of baseband transceiver as shown in Figure 4, it includes Scrambler/Descrambler, Convolution encoder/Viterbi decoder, Puncture/Depuncture, Interleaver/Deinterleaver, FFT/IFFT, GI/ReGI (Guard Interval/Remove Guard Interval) and Mapper/Demapper for modulations of BPSK,QPSK, 16-QAM and 64-QAM. We use 6 Mbps, 12 Mbps, 24 Mbps and 48 Mbps as the information speeds in the designed baseband transceiver. In other words we use the mappings of BPSK, QPSK, 16-QAM and 64-QAM respectively to complete the transceiver system.

The main purpose of the system control circuit is to generate control signals such as timing pulse, enabling line, reset line and selecting line for each OFDM function block. In Figure 4, it depicts the definition of timing

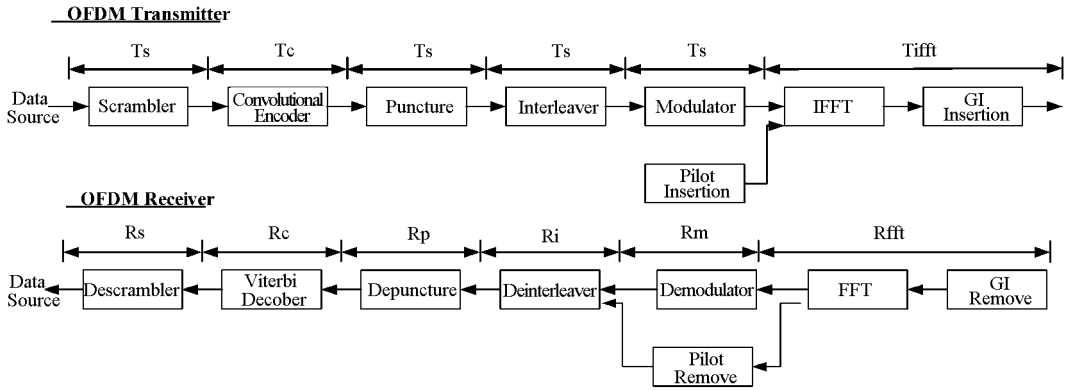

Figure 4. 802.11a baseband transceiver.

pulse required for each transmission speed when the information is transmitted through each block.

The transmission timing pulse is defined as in the following:

(1)The information transmission before IFFT and after FFT: it mainly uses the transmission timing pulses in every 48 information sub-carriers

(2)The information transmission after IFFT and before FFT: it mainly uses the 4μs timing pulses associated with 802.11a OFDM symbol

It mainly uses the system speed as the reference in the design of system timing pulses. It may result in different timing pulses rates in various functional blocks because of their serial or parallel connections. It may also use more than one timing pulses in each functional block. Table 2 lists the timing pulses used in each functional block. Since 20 MHz timing pulse is not related with other timing pulses it is independently generated. By utilizing PLL synthesizing circuits to generate 20 MHz and 144 MHz high speed timing pulses and from these two timing pulses to generate other rate timing pulses for using in other functional circuits. Branching circuit timing pulses are shown in Figure 5.

In OFDM system design, the most complicate part is the timing pulse design for the circuit connecting to FFT/IFFT. The Mapper circuit, which follows by IFFT functional block, has pilot signal inserted and the Mapper operates at 12 MHz, 24 MHz, 48 MHz and 72 MHz rates. It has 48 data sub-carriers and the Mapper output operates

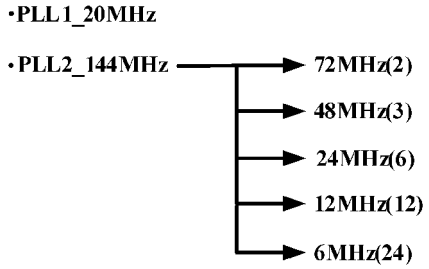

Figure 5. Timing pulses branching trees.

Table 2. System clock design

The clock distribution for Transmitter End								
Modulator	Rate	Code Rate	Ts (MHz)	Tc (MHz)	Tp (MHz)	Ti (MHz)	Tm (MHz)	Tifft (MHz)
BPSK	1101	1/2	6	6	6_12	12	12_20	20
QPSK	0101	1/2	12	12	12_24	24	24_20	20
16-QAM	1001	1/2	24	24	24_48	48	48_20	20
64-QAM	0001	2/3	48	48	48_72	72	72_20	20
The clock distribution for Receiver End								
Modulator	Rate	Code Rate	Rs (MHz)	Rc (MHz)	Rp (MHz)	Ri (MHz)	Rm (MHz)	Rifft (MHz)
BPSK	1101	1/2	6	6	12_6	12	20_12	20
QPSK	0101	1/2	12	12	24_12	24	20_24	20
16-QAM	1001	1/2	24	24	48_24	48	20_48	20
64-QAM	0001	2/3	48	48	72_48	72	20_72	20

at rate 12 MHz, therefore before adding pilot signal each data set of 48 sub-carriers has duration of 4 us. We re-utilize read and write different timing RAM pulses to add 4 sets of pilots and 12 blanks of information. It uses 3 sets for RAM to avoid the possible occurrence of the loss of synchronization signals. It then reads the corresponding information from those required IFFT input positions. Its design is shown in Figure 6. It uses the 3.2μs of the 4μs duration to control IFFT action, to process the 64 sampling points till it reaches the Guard interval to control its circuit action to reach the 4μs of 80 sampling points.

3. Convolutional Encoder & Viterbi Decoder

It has two main categories in channel encoding one is block code and the other is the Convolutional code. Their purpose is to add some information into the input data to enable them having some error correcting capability to correct some errors due to the interference effect in their transmissions. In this paper we consider the Convolutional encoder with (2,1,6) structure as shown in Figure 7. This convolutional encoder has the generating polynomial

$$G(x) = (x^6 + x^4 + x^3 + x + 1, x^6 + x^5 + x^4 + x^3 + 1).$$

Two possible methods are used in the decoding of Convolutional codes it is the sequential decoding and the Viterbi decoding [5−8] as we used in this paper. As shown in Figure 8 the Viterbi decoder consists of 4 basic units:

(a) The Branch Metric Generation Unit (BMU)

When BMU receives new information it will calculate its related branch distances. Its input is binary and 4 branches are formed from 4 phase inverters and adders

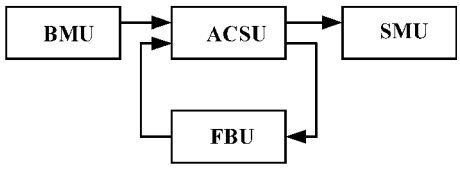

Figure 8. Basic architecture functional blocks of Viterbi decoder.

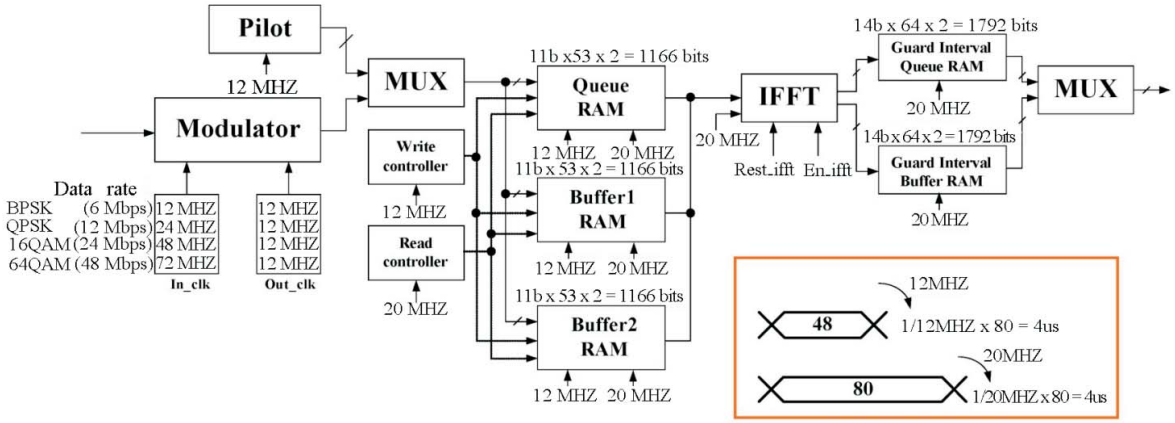

Figure 6. IFFT interface design and timing pulses.

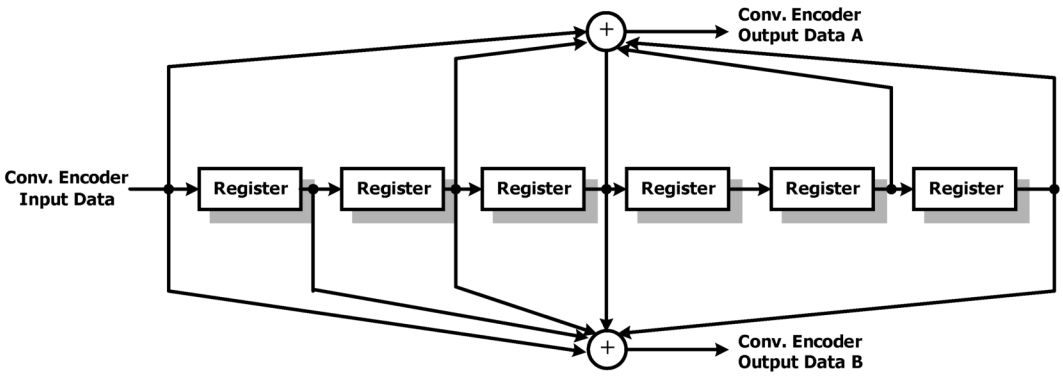

Figure 7. IEEE 802.11a based Convolutional encoder (K = 7).

then. It then calculates and compares the Hamming distances for these 4 branches.

(b) The Add-Compare-Select Unit (ACSU)

The ACSU consists of 64 Processing Units (PE), it continuously executes the addition of path values and branch values and then compares their values to select the branch with the minimum distance as the survival path to be used in SMU.

(c) The Feed Back Unit (FBU)

FBU is used to store every renewed minimum distance path and to detect overflow.

(d) The Survivor Memory Management Unit (SMU)

It uses architecture of trace back to previous stage based on the information of the states in the shift registers and the branch minimum Hamming distances to find the optimum path to recover and decode the data.

4. Radix-2^2 Algorithm in FFT and Hardware Design

Radix-2^2 [9–11] algorithm is modified from radix-4 [12] as shown in Figure 9 [9–11]. The first consideration in the design is that it has 64 FFT data points in wireless area network it is a factor of 4 therefore radix-4 is selected in FFT algorithm. Furthermore in radix-2^2 FFT algorithm it can be constructed as a pipeline architecture and can also be realized with low complicated hardware. By comparing radix-4 and radix-2^2 algorithms they use the same amount of multipliers but it uses the less complexity butterfly architecture in radix-2^2 algorithm it reduces by half in the amount of required adders. By comparing radix-2^2 and radix-2 in hardware implementation it does not need multipliers in the odd stages in radix-2^2 algorithm. Further in the first stage, the third stage and the fifth stage they need only use 1 or –j as their multiplication factor in radix-2^2 algorithm so that it saves a large amount of multipliers in the first 3 odd stages. It is the method to modify the radix-4 algorithm to reduce the hardware complexity.

In FFT design process it has real part and imaginary part in the input and output signals, it uses 11 bits in each set of input signals. Because these input signals pass through A/D converters, quantizers and in order to reduce the possibility of generating errors in the transmission of BPSK, QPSK, 16-QAM and 64-QAM signals we restrict the real and imaginary parts for every input signals within the range from –2 to 2 volts. In Figure 10, it depicts the meaning of each bit.

Also in the design of FFT operation it has many twiddle coefficients we may pay a lot of hardware cost if we use only counters to implement the FFT operation. We select the Altera built-in mega-function ROM and use 12 bits to represent every twiddle coefficients and then use a single control line to read out the required twiddle coefficients. Other memory required is the bit reverse operation. We select Altera built-in random access memory to complete the bit reverse operation. Figure 11 shows the output bits at every stage. It has maximum of 20 bits. Figure 12 [9–11] is the Signal-Path Delay Feedback (SDF) FFT architecture we used for 802.11a based radix-2^2 Decimation in Frequency (DIF).

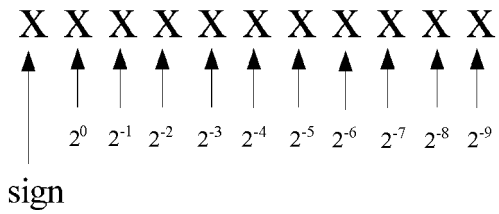

Figure 10. Magnitudes in each bit location.

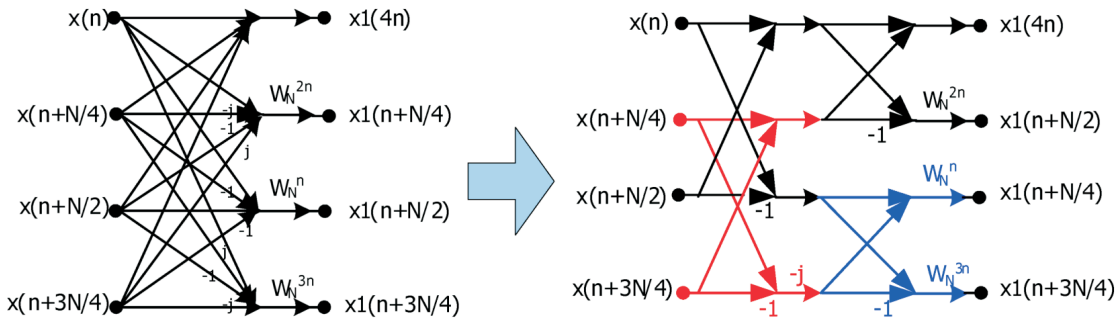

Figure 9. Radix-2^2 algorithm is modified from radix-4.

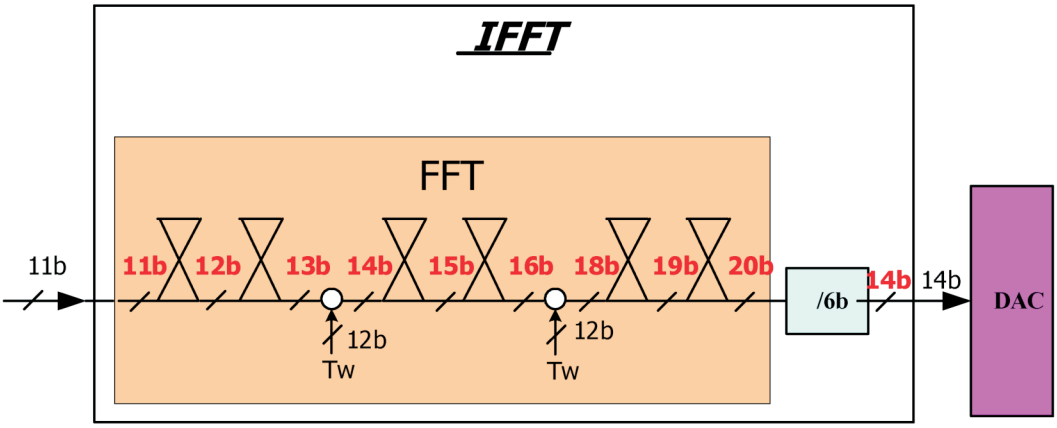

Figure 11. Stages output bits.

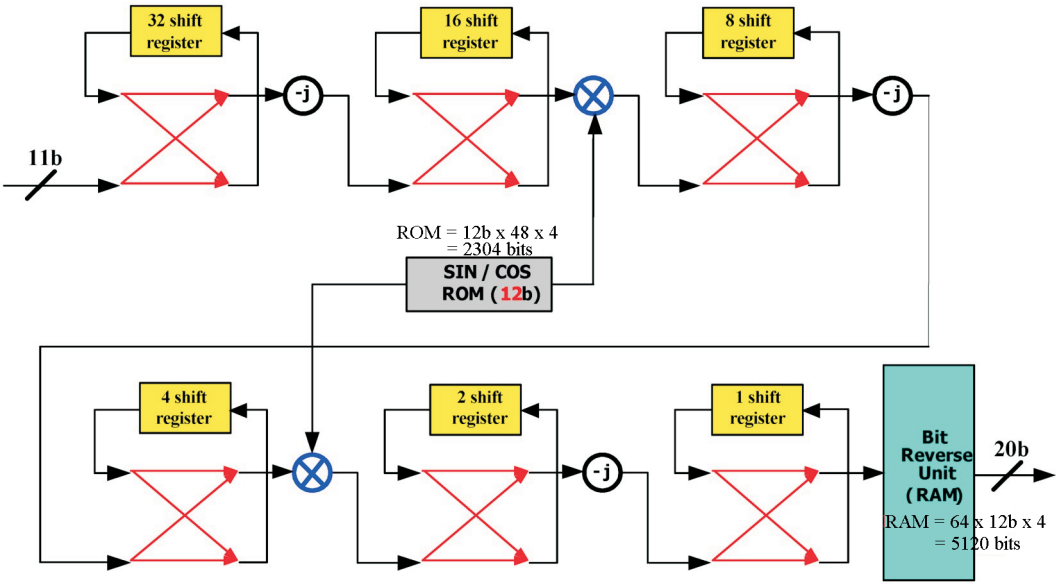

Figure 12. FFT architecture for IEEE 802.11a based system.

5. Simulation Model and Results

5.1 Pathfinder Introduction

We use the Pathfinder developed by Galaxy East Corp Company to validate our designed OFDM baseband transceiver system. The Pathfinder [13] is suitable for the SOC module design. The Pathfinder system we used is the built-in Stratrix EPIS80 FTGA system which can equivalently simulate 470,000 ASIC logic gates. It can reach 7.4 Mbits in the embedded RAM and it has the following merits as fast system integration, short hardware design time, short simulation time, real-time validation and the capability by adding channel noise to test the system performance.

Co-emulation process is shown in Figure 13. After design and encode the whole baseband transceiver system we hand it to MapMaker to generate Pathfinder defined Tcl files. Then the ModelSim combines Wrapper file generated by MapMaker and adds the required test signals to generate Testbench. This Testbench and the synthesized circuit generated from QuartusII are download and burned to Pathfinder. Also we can use the Stratix 80 in the Pathfinder to burn the basband transceiver as the actual circuit and through Pathfinder the Testbench can synthesize real signals to test the baseband transceiver operations. Signals pass through the baseband transceiver can also be feedback to the Modelsim. From these real circuits fabricated we can validate our designed baseband transceiver.

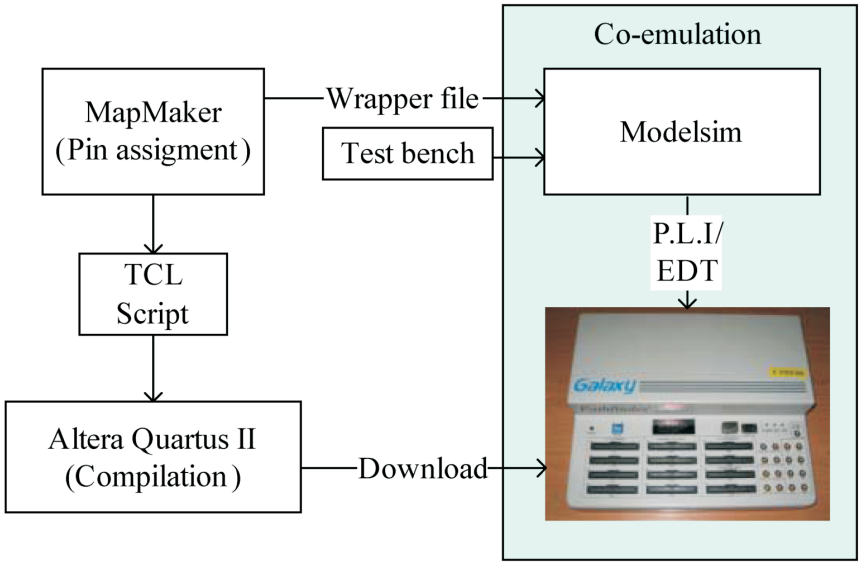

Figure 13. Co-Emulation of Pathfinder and Modelsim.

5.2 Simulation Results

Input signals with data rate of 48 MHz are inputted through the following transmitter blocks: Convolutional encoder, Puncturer, Interleaver, Mapper, IFFT and GI. After passing through GI it connects to ReGI and then pass through various the following receiver blocks such as ReGI, FFT, DeMapper, DeInterleaver, Depuncturer, Viterbi decoder and Descrambler. We then at the Descrambler output to detect the signals received. By comparing the detected received signals and the transmitted signals to check any errors occurred during their transmission. It is observed that it does not have any error encountered in the system simulation as evidenced. It is also observed that the whole system latency is 2249 clocks. Table 3 lists the resulting hardware complexities for each functional block in the designed baseband transceiver.

6. Conclusion

In this paper we used the top-down concept to propose the architecture for a complete baseband transceiver system. In the whole design we use the advantage of fast integration of Pathfinder to reduce the hardware design and simulation time so that it speeds the whole system

Table 3. Resulting functional block complexity in the designed baseband transceiver.

Block	Logic Element & RAM & ROM	Latency (64QAM)	fmax (MHz)
Scrambler	12 LE	1 clocks (48 MHz)	422.12
Descrambler	9 LE	1 clocks (48 MHz)	422.12
Convolutional Encoder	10 LE	1 clocks (48 MHz)	422.12
Viterbi Decoder	9762 LE	74 clocks (48 MHz)	53
Puncture	30 LE	3 clocks (72 MHz)	422.12
Depuncture	33 LE	6 clocks (48 MHz)	422.12
Interleaver	2580 LE & 10112 bits (RAM)	290 clocks (72 MHz)	106.33
Deinterleaver	2353 LE & 10368 bits (RAM)	291 clocks (72 MHz)	106.33
Mapper	873 LE & 23296 bits (RAM) & 660 bits (ROM)	78 clocks (16 MHz)	296.91
Demapper	540 LE & 22528 bits (RAM) & 140 bits (ROM)	108 clocks (12 MHz)	230.95
IFFT	4721 LE & 4096 bits (RAM) & 2752 bits (ROM)	136 clocks (16 MHz)	22.7
FFT	4666 LE & 5632 bits (RAM) & 2752 bits (ROM)	135 clocks (16 MHz)	23.21
Guard Interval	686 LE & 6912 bits (RAM)	82 clocks (20 MHZ)	134.37
Remove Guard Interval	422 LE & 6912 bits (RAM)	67 clocks (16 MHz)	134.37
			Total: 26,697 LE

design process. It also validates the results between the theoretical study and the actual hardware implementations.

Acknowledgement

The authors would like to thank the National Science Council, Taipei, R.O.C. for the financial support under Contracts NSC 93-2745-E-032-001-URD, NSC 93-2213-E-032-023-, and the funding from Tamkang Unversity for the University-Department joint research project.

Reference

[1] Van Nee, R. and Prased, R., "OFDM wireless multimedia communications," Artech House Publishers (2000).

[2] IEEE Std 802.11a-1999 supplement to IEEE standard for information technology — telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher-speed physical layer extension in the 5 GHz band.

[3] IEEE Std 802.11g-2003, Part11: Wireless LAN medium access control (MAC) and physical layer (PHY) Specification, Amendment 4: Further high data rate extension in the 2.4 GHz Band.

[4] Johansson, S., He, S. and Neilsson, P., "Wordlength optimization of a pipelined FFT processor," *Proceedings of IEEE IPPS*, pp. 766–770 (1996).

[5] Omura, J. K., "On the Viterbi decoding algorithm," *IEEE Trans. Inform. Theory*, Vol. IT-15, pp. 177–179 (1969).

[6] Forney, G. D., "The Viterbi algorithm," *Proc. IEEE*, Vol. 61, pp. 268–278 (1973).

[7] Truong, T. K., Shih, M.-T., Reed, I. S. and Satorius, E. H. "A VLSI design for a trace-back Viterbi decode," Communications, *IEEE Transactions* on , Vol. 40, pp. 616–624 (1992).

[8] Byonghyo Shim, et al., "An improved VLSI architecture for Viterbi decoder," *TENCON 99, Proceedings of the IEEE Region 10 Conference*, Vol. 1, pp. 259–262 (1999).

[9] Salivahanan, S., Vallavarj, A. and Gnanapriya, C., "Digital Signal Processing," McGraw Hill (2000).

[10] He, S. and Tokelson, M., "A new Approach to Pipeline FFT processor," *Proceedings of IEEE IPPS,* pp. 766–770 (1996).

[11] He, S. and Tokelson, M., "Desinging pipeline FFT processor for OFDM (de)Modulation," *Proceedings of IEEE URSI Int. Symp. Signals, System, Electron,* pp. 257–262 (1998).

[12] Swartziander, E. E., Young, W. K. W. and Joseph, S. J., "A Radix 4 delay commutator for fast Fourier Transform Processor Implementation," *Solid-State Circuits, IEEE J.* Vol. 19, pp. 702–709 (1984).

[13] Galaxy Far East Corp. Pathfinder™ Data Sheet (2004).

Manuscript Received: May. 5, 2005
Accepted: Jun. 20, 2005