International Journal of Advanced Science and Technology
Vol.97 (2016), pp.69-82
http://dx.doi.org/10.14257/ijast.2016.97.07

Research on Cell ID Blind Detection Algorithm and ASIC Implementation in LTE - A System

Youqi Shen* and Guiyong Li

College of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
syqsnail1128@126.com

Abstract

LTE-A system, the UE and the eNodeB cell synchronization process is very important, it can detect the physical layer cell ID and access to time and frequency synchronization, and then process data communication. The key to cell synchronization is to acquire the cell ID. The ID of the cell can be obtained by ID blind detection, and the cell system information can be obtained. Therefore, it is very important to get the cell ID quickly and accurately. With the constant updating of the protocol, the design of the user's receiving end is also constantly updated. This paper proposes a reasonable and feasible ASIC design scheme and valid UVM (Universal Verification Methodology) verification platform from the perspective of studying the cell ID blind detection algorithm as well hardware implementation by balancing the fast accuracy of data operation with hardware cost, area and power consumption, it has important practical significance. Compared with the same module of other design, ASIC design of this area is 8.8% lower, 9.3% lower power consumption than the other design.

Key words: *Cell ID blind detection; ASIC implementation; UVM verification*

1. Introduction

With the LTE-A system of large-scale commercial, for different emerging markets, the user design of the receiver is constantly updated. In the cell synchronization process, the UE and the eNodeB are obtained to get time and frequency synchronization and cell ID, which are mainly divided into primary synchronization signal (PSS) detection and secondary synchronization signal (SSS) detection [1]. For the PSS detection algorithm, literature [2] gives an algorithm with low computational complexity. The algorithm first obtains the position of the PSS according to the symmetry of the PSS, and then cross-correlates with the local PSS to obtain the ID number. For the SSS detection algorithm, literature [3] compared the SSS detection algorithm in time and frequency domain, through simulation, SSS detection algorithm performance is relatively better in the frequency domain. In literature [4, 5], an algorithm using Hadamar matrix instead of correlation algorithm is proposed, which can reduce the complexity of SSS detection algorithm.

In this paper, through the algorithm of blind detection of cell ID, ASIC design is compared with the traditional Field Programmable Gate Array (FPGA) processing, the ASIC design is more suitable for real-time and reliability requirements of large-scale and complex digital signal processing, cost, power consumption, etc. At the same time, through the verification of hardware and algorithm performance, the design module has better performance in area and power consumption than the same kind of design module through the verification of UVM. So it can meet and apply to LTE-A user terminal and has certain practical significance.

ISSN: 2005-4238 IJAST

2. Related Algorithms Research

2.1 PSS Coarse Synchronization

PSS coarse synchronization is to quickly find the approximate location of PSS and get the cell group ID number $N_{ID}^{(2)}$. Since the PSS has a period of 5ms and each field contains PSS and the position is fixed, it is possible to take the data in two consecutive fields and slide the correlation data between the two sets of received data. The position of the maximum value is the approximate position of PSS, The received PSS is associated with the local PSS according to the approximate location of the PSS to obtain the $N_{ID}^{(2)}$.

The specific steps of the program are: receive a frame of data, divided into two fields, respectively, using $r_1(n)$ and $r_2(n)$ to represent (assuming that each field data contains a complete PSS), the two sets of receive data. And the correlation function is given by:

$$C(d)=\sum_{n=0}^{N-1} r_1(d+n)r_2^*(d+n) \tag{1}$$

Where N denotes the length of the correlation window, that is, an OFDM symbol length. When the down sampling rate is set to 1/16, N takes 128; d indicates the sliding distance, and the sliding range is $[0, 9471]$. The average energy of the two groups of received data is given by:

$$R(d)=\frac{1}{2}\sum_{n=0}^{N-1}(|r_1(d+n)|^2+|r_2(d+n)|^2) \tag{2}$$

The timing measure function is:

$$TM(d)=\frac{|C(d)|^2}{|R(d)|^2} \tag{3}$$

The approximate position of PSS can be obtained as:

$$\hat{d}=\arg\max_d\{TM(d)\} \tag{4}$$

2.2 PSS Fine Synchronization

PSS coarse synchronization is only to obtain the approximate position of PSS, the accuracy position cannot meet the requirements, need to carry out PSS fine synchronous operation. In this paper, the algorithm based on receiving PSS and local PSS is adopted.

First, an $N_{ID}^{(2)}$ obtained by coarse synchronization of the PSS is locally generated in the frequency domain PSS, and then transformed into the time domain by the IFFT. Then a series of values are obtained by sliding correlation calculation in the range of 64 points around $\hat{d}$, *i.e.*, $[\hat{d}*16-64, \hat{d}*16+63]$, and then the time-domain received data before descending. The position where the maximum value is located is PSS fine synchronous position. The correlation function is as follows:

$$C(d)=\sum_{n=\hat{d}*16-64}^{\hat{d}*16+63}\left|r(d+n)s^*(n)\right|^2 \tag{5}$$

The position of the maximum value obtained by Eq. (5) is the position of the fine synchronization:

$$\hat{d}_{PSS} = \arg\max_{d}\{C(d)\} \tag{6}$$

2.3 SSS Detections

After the PSS synchronization is completed, the exact position of the PSS is determined. However, it cannot be determined whether the current data is the first or second field. The SSS takes 10ms as the cycle, so it can detect the cell group ID $N_{ID}^{(1)}$. In order to reduce the complexity and reduce the computational complexity, the parameters m_0 and m_1 are obtained by descrambling. According to the one-to-one correspondence between m_0 and m_1 and $N_{ID}^{(1)}$, the cell ID group number $N_{ID}^{(1)}$. In this paper, we use the coherent detection algorithm [6].

Since the SSS detection is performed after the PSS detection, when the channel coherence time is greater than 4 OFDM symbol lengths, the PSS is used to obtain the estimated value $\hat{H}_{PSS}(k)$ of the channel impulse response, and the frame synchronization is performed by the coherent detection method.

The received time domain PSS is transformed into the frequency domain PSS by the FFT transformation, and is denoted by $R_{PSS}(k)$, and the local frequency domain PSS is generated, denoted by $T_{PSS}(k)$. If the channel coherence time is greater than 4 OFDM symbols, the channel impulse response estimate can be obtained:

$$\hat{H}_{PSS}(k) = R_{PSS}(k) / T_{PSS}(k) \tag{7}$$

According to the PSS position, the time-domain SSS is transformed into the frequency-domain SSS by FFT, and then the channel compensation is performed:

$$\hat{R}_{SSS}(k) = R_{SSS}(k)\hat{H}_{PSS}(k) \tag{8}$$

And divides $\hat{R}_{SSS}(k)$ into a sequence $\hat{R}_{SSS}(2k)$, $\hat{R}_{SSS}(2k+1)$ consisting of even and odd-numbered bits. According to the PSS synchronization obtained by the $N_{ID}^{(2)}$ and scrambling sequence $c_0(k)$, the $\hat{R}_{SSS}(2k)$ to descramble:

$$a_{m0}(k) = \hat{R}_{SSS}(2k)c_0(k) \tag{9}$$

The different cyclic shift sequences $S^{(i)}(k)$ of $a_{m0}(k)$ and m sequences are segmented and the estimated value m_0 is obtained according to the maximum value $\hat{m}_0$:

$$\hat{m}_0 = \arg\max_{i}\{\sum_{j=0}^{M-1}|\sum_{k=jN_M}^{(j+1)N_M} a_{m_0}(k)S^{(i)}(k)|^2\} \tag{10}$$

The above formula $i = 0,1,...,30$, M is the number of segments, N_M is the data length, the above formula $M = 4$.

Generate a local scrambling code $c_1(k)$ and $z_1^{(\hat{m}_0)}(k)$ two sequences, the $\hat{R}_{SSS}(2k+1)$ descrambling operation, too:

$$a_{m1}(k) = \hat{R}_{SSS}(2k+1)c_1(k)z_1^{(\hat{m}_0)}(k) \tag{11}$$

m_1 estimate of $\hat{m}_1$ and $\hat{m}_0$ similar, not repeat them:

$$\hat{m}_1 = \arg\max_i \{\sum_{j=0}^{M-1} | \sum_{k=jN_M}^{(j+1)N_M} a_{m1}(k)S^{(i)}(k)|^2\} \tag{12}$$

The correspondence between m_1 and m_0 estimates $\hat{m}_0$ is as follows:

$$\begin{cases} m_1 \in [0,1,\cdots,\hat{m}_0+7], & \text{when } \hat{m}_0 \in [0,\cdots,2], \text{ and } m_1 \neq \hat{m}_0 \\ m_1 \in [0,1,\cdots,\hat{m}_0+6], & \text{when } \hat{m}_0 \in [3,\cdots,7], \text{ and } m_1 \neq \hat{m}_0 \\ m_1 \in [\hat{m}_0-7,\cdots,\hat{m}_0+6], & \text{when } \hat{m}_0 \in [8,9], \text{ and } m_1 \neq \hat{m}_0 \\ m_1 \in [\hat{m}_0-6,\cdots,\hat{m}_0+6], & \text{when } \hat{m}_0 \in [10,\cdots 24], \text{ and } m_1 \neq \hat{m}_0 \\ m_1 \in [\hat{m}_0-6,\cdots,30], & \text{when } \hat{m}_0 \in [25,\cdots 30], \text{ and } m_1 \neq \hat{m}_0 \end{cases} \tag{13}$$

Can be obtained, $\hat{m}_0$ is less than $\hat{m}_1$, the receiving SSS is in subframe 0. Otherwise in the subframe 5.

Let $M = \hat{m}_0 - \hat{m}_1$, then $N_{ID}^{(1)}$ can be obtained by the following formula:

$$N_{ID}^{(1)} = 30(M-1) + \hat{m}_0 - (M-1)(M-2)/2 \tag{14}$$

Using the descrambling coherent detection algorithm, a total of 44 correlation operations (estimated m_0, 30 times; estimate m_1, 14 times).

3. ASIC Design and Implementation

3.1 Functional Structure

The cell synchronization ID blind detection module includes a control module (ctrl), an interface module (regif), a memory module (mem) and an arithmetic processing module (cal). The external interaction with AXI Direct memory access (AXIDMA) triggers the reading and writing of Double Data Rate (DDR) data.

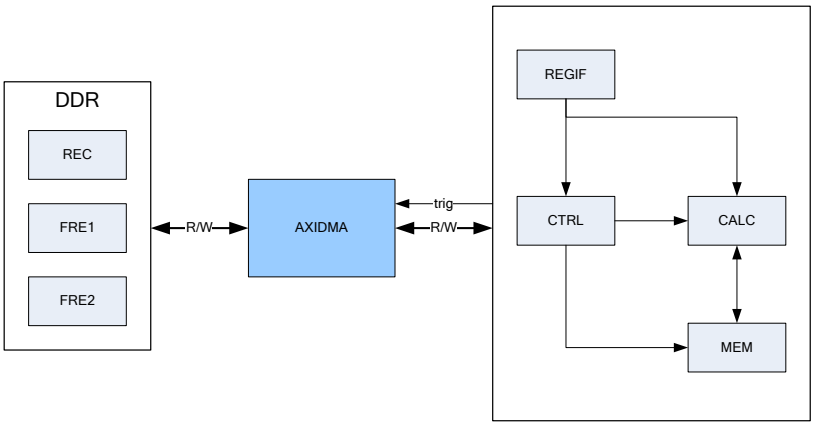

Figure 1. Block Diagram of the Functional Block Structure

3.1.1. Control Module

In this paper, the control module with finite state machine control, the jump as follows:

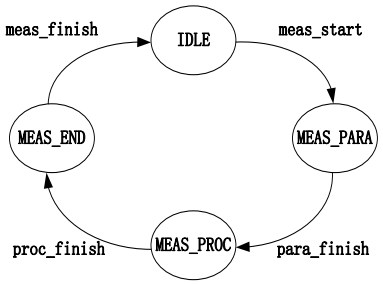

Figure 2. Module State Machine Control Transfer Diagram

1. IDLE state: not activated are in this state, when meas_start is valid, IDLE state jumps to MEAS_PARA state.

2. MEAS_PARA state: After entering this state, when the register parameter is read, para_finish is set to 1, indicating that the parameter reading is complete and the MEAS_PARA state jumps to the MEAS_PROC state.

3. MEAS_PROC state: enter the state, the specific functions of the module operation. When the operation is complete, the proc_finish signal is set to indicate that the operation has been completed and the MEAS_PROC state jumps to the MEAS_END state.

4. MEAS_END state: When the module enters this state, it indicates that the external configuration of the functional tasks have been completed, meas_finish signal is set to jump from the MEAS_END state to IDLE state.

3.1.2. Interface Module

The interface module uses the common ZSP bus interface, sets up the parameter which needs to the module correlation function.

Table 1. Interface Signal List

Signal name	Direction	Description
Clock and reset signal		
clk	Input	Function clock
rst_	Input	Function reset, active low
zclk	Input	Interface clock
zrst_	Input	Interface reset, active low
ZSP interface signals		
zcs_	Input	ZSP bus chip select, active-low
ziosel	Input	ZSP bus register selection, high-level access to the register, low-level access to memory
zwr_[1:0]	Input	ZSP bus write enable, active low；zwr_[0] is low：Low 32-bit write enable；zwr_[1] is low：High 32-bit write enable；zwr_[1:0] is low：64-bit write enable
zrd_	Input	ZSP bus read enable, active low
zaddr[11:0]	Input	ZSP bus address
zrdata[63:0]	Output	ZSP bus read data
zwdata[63:0]	Input	ZSP bus write data
Other signals		
dma_start	Input	Synchronous with the interface clock sclk, the rising edge of the hardware module.
fft_irq	Output	Interrupt signal, active-high. Synchronized with the interface clock sclk, high level to maintain 32 zclk clock cycle.
clk_en	Input	Memory clock enable signal. When the module is idle and the external memory is not accessed, pulling this signal low can reduce the module memory power consumption.

Interface module in addition to the need to design interface signals, but also need to design parameter registers, easy ZSP bus to read and write, to achieve different functional module configuration.

Table 2. Module Register List

Name	Description
MEAS_FFF_CTRL	Control register
FFT_PARA	FFT parameter register
FRE_PARA	Frequency offset compensation parameter register
GEN_PARA	Produces a local PSS or SSS parameter register
MAX_PARA	Maximum value lookup parameter register
PSS_H_PARA	PSS impulse response calculation parameter register
M0_PARA	M0 Value estimation parameter register
M1_PARA	M1 Value estimation parameter register
MEAS_FFF_INTF	Interrupt Flag Register
FFT_MAX_GENE	Maximum and normalization factor registers

3.1.3. Memory Module

In this paper, the design of the module, a total of 8 memory, each memory size are Spram128x64bw, both as input and output memory. According to the different functions of the module, the specific allocation of the use of eight blocks memory.

3.1.4. Arithmetic Processing Module

This module mainly implements PSS coarse synchronization, PSS fine synchronization, SSS blind detection in the computing processing unit. As follows:

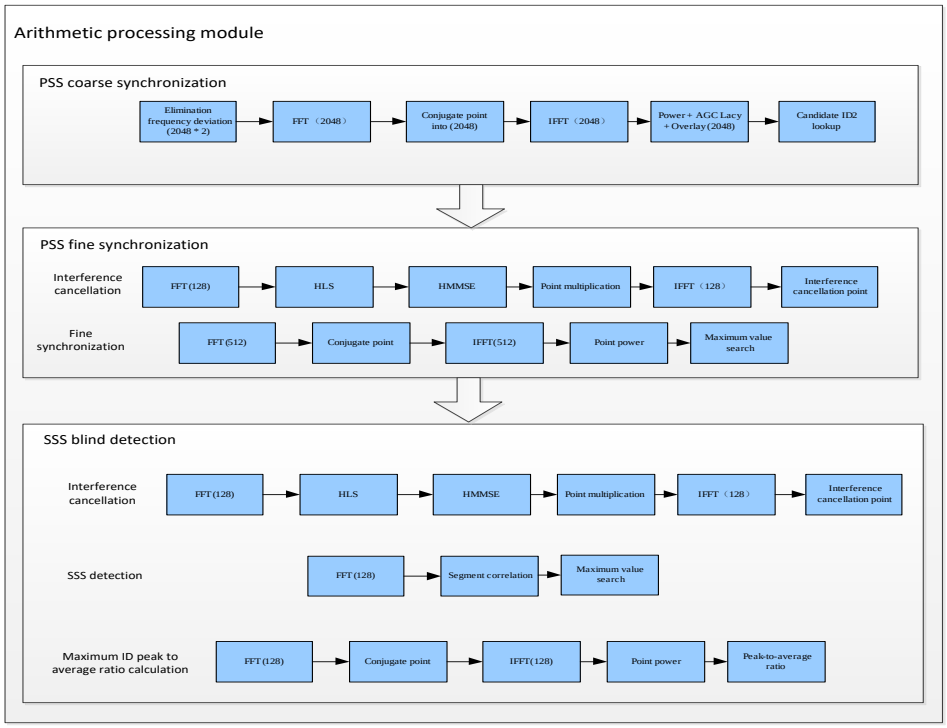

Figure 3. Module Operation Processing Unit Flow Chart

3.2 Hardware Implementation Process

ID blind detection hardware implementation steps and the corresponding storage direction as shown below:

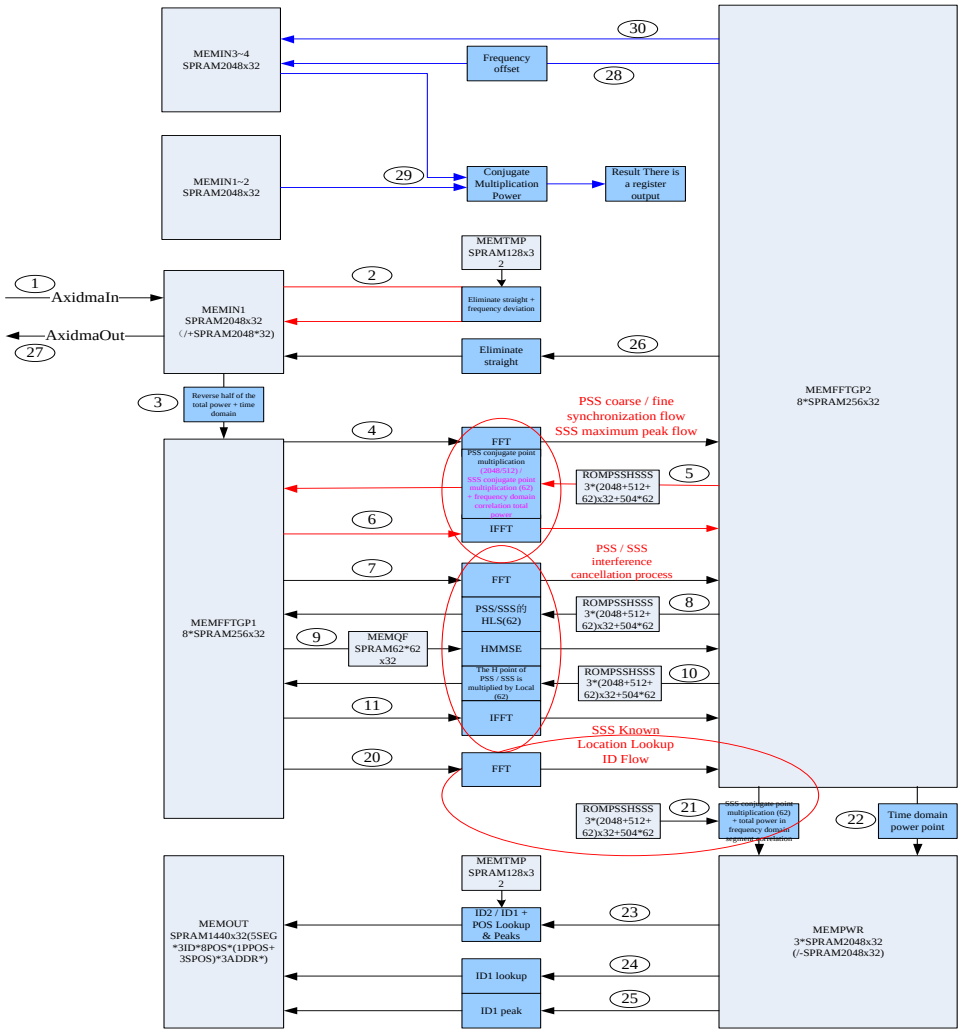

Figure 4. Module Hardware Implementation Flow Diagram

PSS coarse synchronization: steps 1, 2, 27, 3, 4, 5, 6, 22, 23;
PSS fine synchronization: steps 1, 3, 4, 5, 6, 22, 23;
PSS interference cancellation: steps 1, 3, 7, 8, 9, 10, 11, 26, 27;
PSS known $N_{ID}^{(2)}$ **coarse synchronization: steps 1, 2, 27, 3, 4, 5, 6, 22, 23;**
SSS known position finding $N_{ID}^{(1)}$**: steps 1, 3, 20, 21, 24;**
SSS maximum $N_{ID}^{(1)}$ **peak average calculation: steps 1, 3, 4, 5, 6, 22, 25;**
SSS interference cancellation: steps 1, 3, 7, 8, 9, 10, 11, 26, 27;
SSS known position and ID pair ID order: steps 1, 3, 20, 21, 24;

3.3 RTL Code to Achieve Results

According to the above algorithm research and hardware implementation flow, Resistor Transistor Logic (RTL) code is written, the ASIC implementation of the module

is completed. After the code is compiled, it is checked by VCS simulation tool for its correctness and feasibility [7]. The code inspection report is shown below.

```
                    ALL ERROR STATISTICS
================================================================
================================================================
                    Total Error Count = 0

                    ALL WARNING STATISTICS
================================================================
LEDA                        B_3200                      26
LEDA                        B_3212                       2
================================================================
                  Total Warning Count = 28
```

Figure 5. RTL Code Inspection Report

In this paper, the use of Verdi simulation tool RTL code for the module was integrated, this will help to check the code whether it can be integrated and normative, integrated top-level structure shown below. The top-layer structure mainly comprises a control module meas_fft_ctrl, an interface module meas_fft_regif, a memory module meas_fft_mem, a butterfly operation unit meas_fft_proc, a frame synchronization and a cell group ID number detection function module meas_fft_sss, and a frequency offset compensation function module meas_fft_fre.

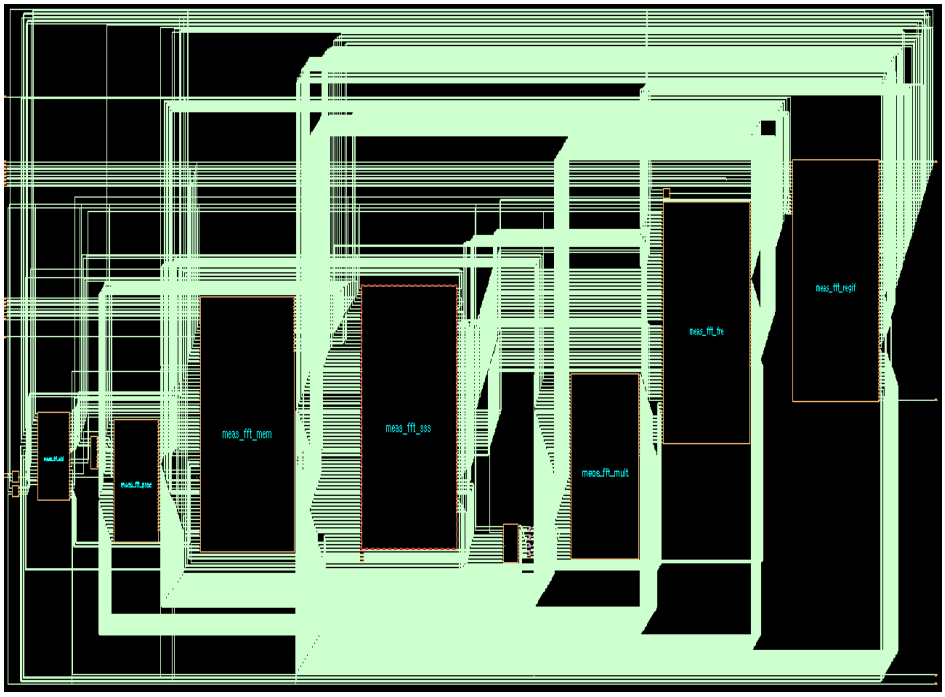

Figure 6. Top Layer Circuit Diagram

RTL code designed to achieve the function of the target, but also the need for functional verification.

4. UVM Simulation Verification

The verification platform of ID-based blind inspection module is mainly based on the UVM authentication method of SystemVerilog verification language, which mainly consists of two parts: random verification environment and reference model [8].

The block diagram of the entire verification platform for the module is shown below.

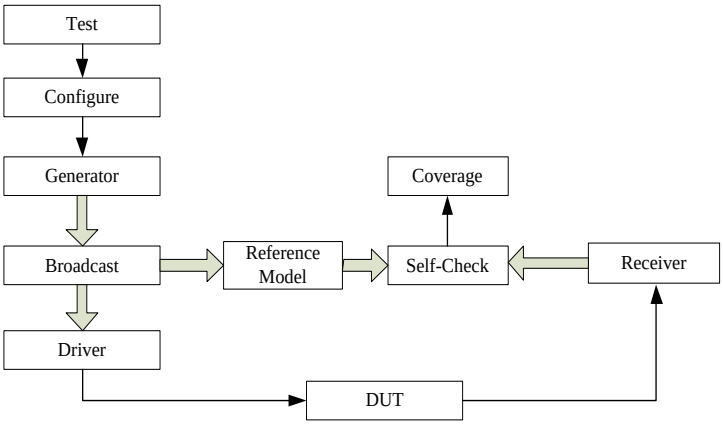

Figure 7. Module Verification Platform Structure

4.1 Simulation and Verification Results

This section use VCS simulation tools and graphical debugging tools, the module was a comprehensive functional verification, the verification process need to cover all the function points, simulation waveform more, so we select several typical waveform analysis instructions.

4.1.1. Register Basic Attribute Simulation Results

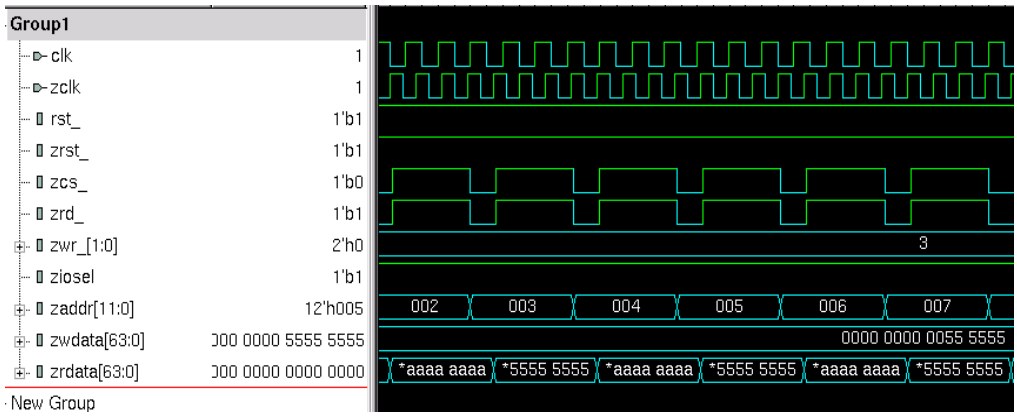

Figure 8. Interface Timing and Write Register Waveform

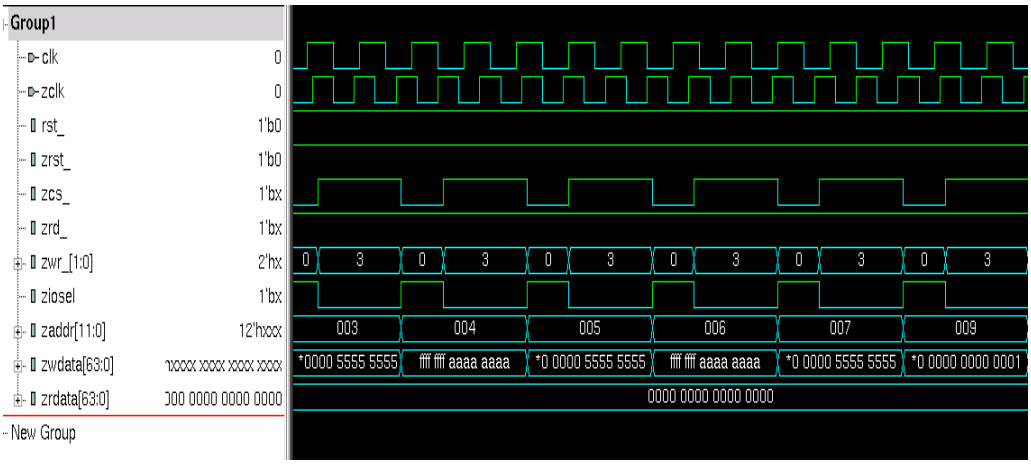

Figure 9. Register Reads the Waveform

It can be seen from the figure, the module to meet the bus interface timing; the module's registers can be normal reset, and the reset value is correct; the module's registers can read and write, and read and write attributes fully meet the design intent.

4.1.2. M0 Value Estimation Function Simulation Results

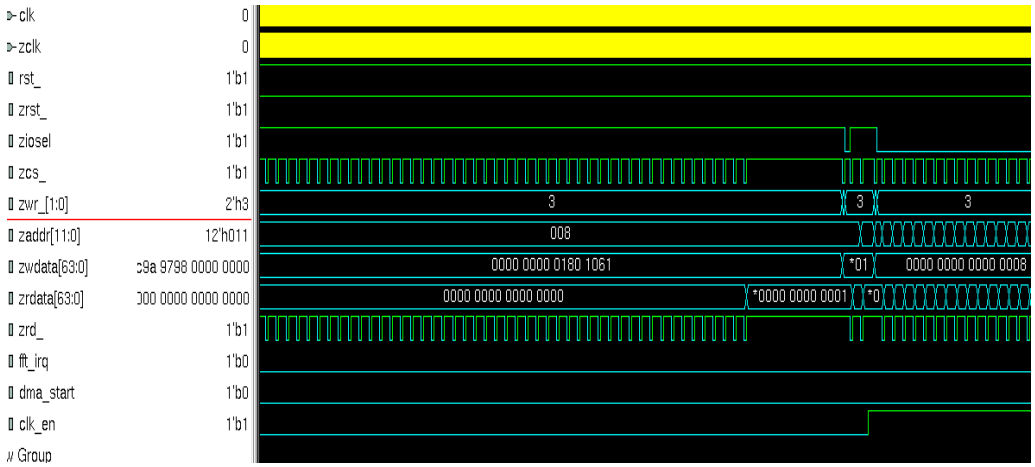

Figure 10. M0 Value Estimation Function Simulation Waveform

As can be seen from the figure, this simulation does not enable interrupts, so after the end of the module operation, and no interrupt signal. In this case, the value of the interrupt flag register can only be read until the interrupt flag register is set to determine that the module has finished running, and then read the output data from the memory and compare the correctness of the output data. The results show that this function can achieve the desired goal.

4.1.3. M1 Value Estimation Function Simulation Results

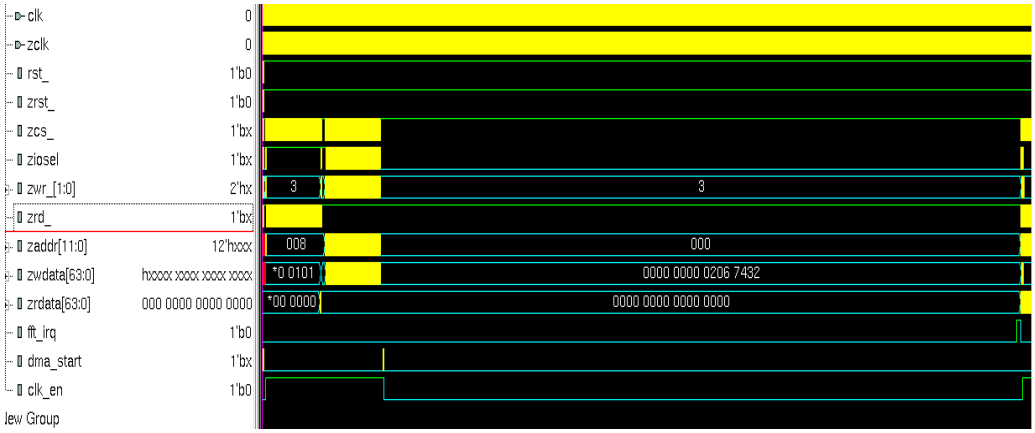

Figure 11. M1 Value Estimation Function Simulation Waveform

Figure 11 shows the simulation waveform for the M1 value estimation function. From the waveform can be seen, this simulation is by the DMA boot module work. After the module runs, the interrupt signal is normally generated. Comparison of the data after the simulation shows that the function can achieve the desired goal.

4.2 Simulation Results Analysis

By observing the simulation waveforms and emulation log information of each test case, it can be concluded that the registers and the memory can be read and written normally, and there is no data bit sticking or address error, *etc*. The bus operation is normal and there is no bus which cannot be read and written. The actual working time of the module is shown in Table 3, and the actual working time is in accordance with the time evaluation in the program; the interrupt can be generated normally; each function of the module can achieve the expected target.

```
Command: ./simv -lca -cm line+fsm+cond+tgl +vmm_log_nofatal_at_1000=NORMAL +vmm_log_nowarn_at_20
ed_automatic
Chronologic VCS simulator copyright 1991-2012
Contains Synopsys proprietary information.
Compiler version G-2012.09; Runtime version G-2012.09;  Feb 20 20:07 2014
NOTE: automatic random seed used: 1518744149
VCD+ Writer G-2012.09 Copyright (c) 1991-2012 by Synopsys Inc.

[Test Configuration] simulation_cycle = 12 simulation_zcycle = 8 wr_type = 1
[Test Configuration] M1_EN = 0 M0_EN = 0 RESS_MODE = 0 RESS_EN = 0 SSS_CORRE_EN = 0
[Test Configuration] MAX_EN = 0 SL_CORRE_EN = 0 MODE_SSS = 1 PSS_SSS_OUTSEL = 0
[Test Configuration] PSS_SSS_EN = 1 LNUM_MOD1 = 13 FRE_SEL = 0 FRE_EN = 0 CLEAR_EN = 0
[Test Configuration] FFT_INTEN = 1 FFT_IFFT_SEL = 0 FFT_CAL = 0 DMA_START_EN = 0 FFT_START = 1
[Test Configuration] FRE_IN = 0 FFT_NPTS = 0 INOUT_CTRL = 0 OUT_REV = 0
[Test Configuration] Senario = SSS_GEN
[Test Configuration] SFNUM = 1 NID2 = 1 NID1 = 132 sss_gen_choice = 0

Normal[NOTE] on Environment(Environment) at      0.0ns:
    *********the reset is on *********
At 1544.0ns:+++++Irq go up+++++
At 1800.0ns:+++++Irq go down+++++


Normal[NOTE] on Scoreboard(sb) at   2928.0ns:
    Successfully Compare

Normal[NOTE] on Scoreboard(sb) at   2928.0ns:
    1 packets checked.  Coverage = 2.466898
Simulation PASSED on /./ (/./) at   2928.0ns (0 warnings, 0 demoted errors & 0 demoted warnings)
irq_count=1
$finish at simulation time   2928.0ns
"/work/c8320chip/verification/guox/C8320_MEAS_FFT_ML/env/coverage/meas_fft_sva.sv", 35: meas_fft

------------------------------------------------------------------------
VCS Coverage Metrics: during simulation line, cond, FSM, tgl was monitored
------------------------------------------------------------------------
          V C S   S i m u l a t i o n   R e p o r t
```

Figure 12. Module Emulation Log Information

Table 3. Lists the Module Evaluation Time Versus the Actual Working Time

	Evaluation time	real time
FFT (2048 points)	3000 cycles	2934 cycles
Generates a local PSS	384 cycles	384 cycles
A local SSS is generated	384 cycles	384 cycles
Maximum value search	9544 cycles	9329 cycles
M0 value estimation	768 cycles	683 cycles
M1 value estimation	896 cycles	873 cycles

4.3. Logic Synthesis

In this paper, the use of DC synthesis tools on the logic of the module synthesis, it can RTL code into the gate-level netlist, and produce the appropriate logical file reports.

In order to effectively reduce power consumption and reduce the chip area, this paper uses a pre-insert with the scanning side of the latch type integrated gate clock. The results are shown below.

```
39 Combinational area:          299669.322752
40 Buf/Inv area:                35954.754388
41 Noncombinational area:       1002722.948914
42
43 Total cell area:             1302392.271666
```

Figure 13. Module Area Report

```
46    Cell Internal Power  =   19.3219 mW   (38%)
47    Net Switching Power  =   31.9527 mW   (62%)
48                           ---------
49 Total Dynamic Power     =   51.2747 mW  (100%)
50
51 Cell Leakage Power      =   16.4480 uW
```

Figure 14. Module Power Report

The following conclusions can be drawn from Figure 13 and Figure 14:

1, the design of the integrated logic area is 1 302 392.271 666 um2, where the combinational logic area is 299 699.322 752 um2, the sequential logic area is 1 002 722.948 914 um2;

2, the design of the operating voltage is 0.99, dynamic power consumption is 51.274 7mw, the static power consumption is 16.448uw, the total power consumption is 51.291 1mw.

5. Conclusion

The logical area of the synchronous ID blind detection module in LTE-A system designed by this paper is 125 669.429 722 um2, about 8.8% smaller than that of the same module. The power consumption of the module is 5.259 2 mw, about 9.3% lower than that of the same module. It can be seen, in line with this without sacrificing the premise of the larger performance of the hardware area, power consumption considerations, compared with similar modules designed to have certain advantages, to meet the design requirements, with application value.

References

[1] A. Andreescu, A. Ghita and A. A. Enescu, “Long term evolution primary synchronization algorithms”, Electronics and Telecommunications (ISETC), 2010 9th International Symposium on. IEEE, **(2010)**, pp. 125-128.
[2] Z. Zhang, J. Liu and K. Long, “Low-complexity cell search with fast PSS identification in LTE”, Vehicular Technology, IEEE Transactions, vol. 61, no. 4, **(2012)**, pp. 1719-1729.
[3] J. Zhang and X. Huang, “Autocorrelation based coarse timing with differential normalization”, Wireless Communications, IEEE Transactions, vol. 11, no. 2, **(2012)**, pp. 526-530.
[4] J. I. Kim, J. S. Han and H. J. Roh, “SSS detection method for initial cell search in 3GPP LTE FDD/TDD dual mode receiver”, Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on. IEEE, **(2009)**, pp. 199-203.
[5] A. Andreescu, A. Ghita and A. A. Enescu, “Long term evolution primary synchronization algorithms”, Electronics and Telecommunications (ISETC), 2010 9th International Symposium on IEEE, **(2010)**, pp. 125-128.
[6] Y. Sheng and X. Luo, “Algorithm Study on Cell Search in LTE”, Communications Technology, vol. 3, **(2009)**, pp. 035.
[7] Synopsys. VCS User Guide Version 6.1. 2014: 12-18.
[8] M. Keaveney, A. McMahon and N. O'Keeffe, “The development of advanced verification environments using system Verilog”, Signals and Systems Conference, 208. (ISSC 2008). IET Irish. IET, **(2008)**, pp. 325-330.

Author

Youqi Shen, got his B.S. degree from Weifang University, and his Master degree from Chongqing University of Posts and Telecommunications, year in 2014 and now. His main research is mobile communication terminal technology.