



Smart Coherence for SOC Test

Jinlei Liu

1 Preface

Coherence is a state or situation in which all the parts or ideas fit together well so that they form a united whole. The term of coherence was originally found in optics from an experiment called double-slit experiment by Thomas Young in the early of 1800. Young's experiment demonstrated a beautiful phenomenon of light when passing through two slits to interfere, producing bright and dark bands on the screen. Ever since then the term "coherence" has been used in any field that involves waves. In electrical engineering, electrical signal is described as wave too with amplitude and frequency. In this article, we are talking about an SOC tester at which various components such as digital channels, their frequencies rates as well as data exchange are put together to work in an elegant and coherent way.

2 Introduction

In the internet era, integrated circuit performance doubles in almost every 18months, the Moore's Law. With the introduction of SmartPhone led by Apple, phone is no more a phone only to make a phone call. More often people use the device to send e-mail and text message and even more and more to send or share audio and video across the global. The enabler of this enormous success is various units inside a SmartPhone running at different rates. Fig. 1 shows an overview on one of design example of a SmartPhone.

There are processors such as Application Processor and Communication Processor or base-band processor. The application processor alone could hold multiple special processors for video and audio data processing. They are processing audio and video data in massive manner. There are multiple interfaces to transfer data to and from the external media such as hard disk, memory. There are multiple buses running inside the chip such as Display Port, Camera Port, Memory Port, also USB etc. All these components collaborate with each other in the harmonic and elegant manner to produce the beautiful video, voice and various applications on these devices.

With the technology evolving, integration of these interfaces into a single SOC chip is the trend, hence also expensive and complicated testing equipment to test these devices. Like these elegantly operating devices, the testing equipment must be "elegantly" designed of being capable of concerting with the operations of such devices.

3 Coherent SOC Test System

A simplest form of a coherent SOC test system refers to its clock architecture, in which clocks of all resources are referenced to the same clock source but will generate the needed clock in a coherent way according to the equation:

$$F_1/F_2=M/N,$$

(1)

where F_1 and F_2 are frequencies, and M and N are integer numbers.

Fig. 2 shows how this could look like.

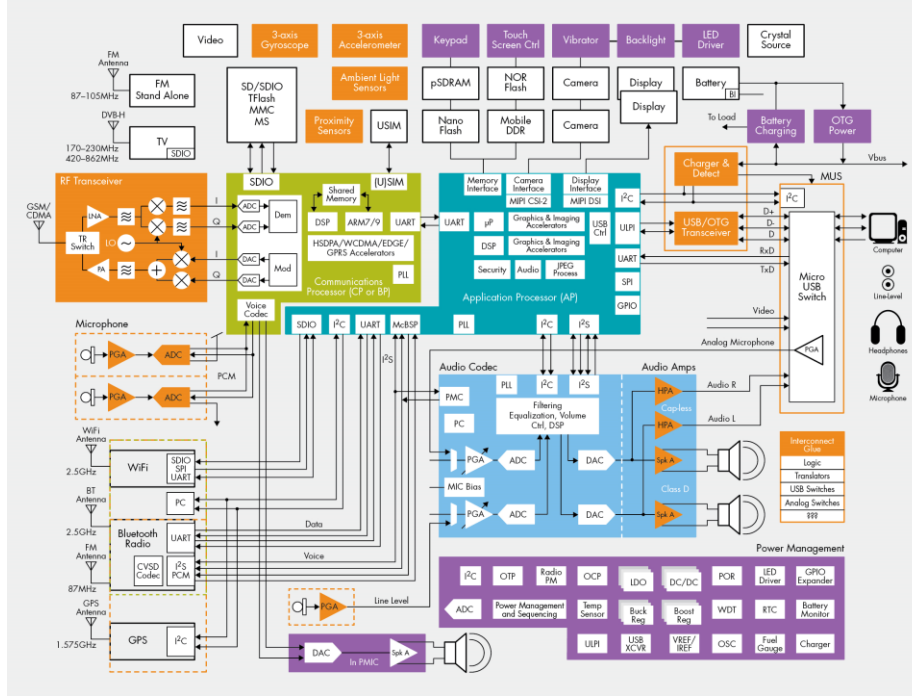


Fig. 1: Typical SmartPhone Block, source: http://www.eetimes.com/document.asp?doc_id=1278458

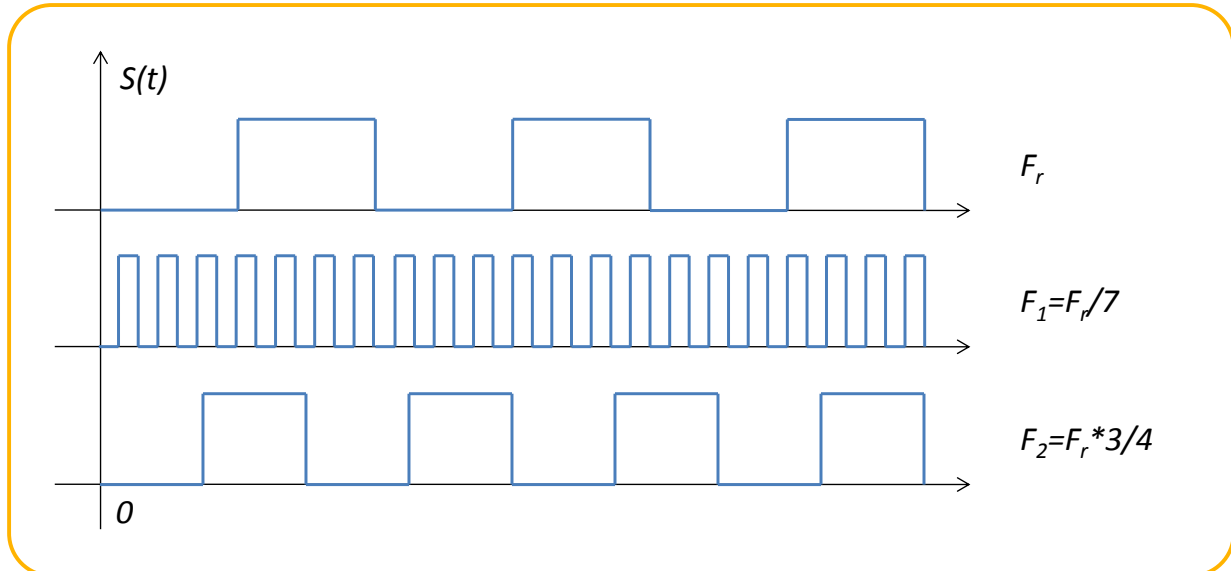


Fig. 2: Coherent Clock System

Clock F_1 and F_2 are derived from a common reference clock source F_r . The beauty of the clock generation in this manner is that it removes away any accuracy issue associated with the clock. It mimics the similar way (when not exact) as the device operates in the real application, where the center piece of the device is a PLL through which various clocks are generated. In most cases, device clocks follow the M over N

ratio. Although there are exceptions, in almost all cases IC designers do allow the clock generation follows the M over N notation, thus makes the test on such an SOC test system an easy task.

V93000 clock architecture follows the coherent equation in both its PinScale system and the latest SmarScale system. All of its data channel clock periods are calculated based upon the coherent equation shown before and generated also in such manner. From now on, it is important to keep ratio of all different clocks precise in the M over N notation, the exact value of the clock is then in the secondary order. Why? In any of such an SOC chip, as indicated before, there is always a PLL. The PLL always needs a reference clock to its input and all other clocks inside the chip are derived from this reference clock using M over N coherent equation. The reference clock can be a crystal or a signal from a tester channel. If clocks of all tester channels are also precisely generated in term of M over N coherent equation to this reference clock, then all clocks are interlocked to the correspondent device clock too. There will be no long-term out of phase shifting. Consequently, it is no more of any importance to create precise clocks in term of its numeric value. Rather it is absolutely imperative that each of these clocks is interlocked precisely in term of the coherent equation. Coherence is also critical in testing RF device. Often times due to the reason of one or another, desired frequency can't be generated precisely and thus it results in the frequency slightly offset to the frequency to be. If this is the case, the consequence will be disastrous.

In this article, different use cases will be presented. First we will present a method of using the coherent sampling technique to construct high speed analog waveform using just digital channel. Second we will describe how we use V93000 protocol aware solution to test a complex RF device. And third we will present a brand new method of doing the test without physical pattern over the internet.

4 Coherent Waveform Reconstruction

In this section, we describe how a coherent SOC system can be used to reconstruct an analog waveform. Conventionally when using a digital channel to reconstruct the waveform such as scope tool in timing diagram, both timing and level are being moved correspondently. Inherently this technique tends to take a long execution time and the achievable timing resolution is limited too. The method of the coherent waveform reconstruction utilizes the coherency of V93000 SOC system to reconstruct the waveform of a periodic signal. It uses the coherent sampling technique in mixed-signal application. Let's refresh our memory on how an analog waveform is being captured in mixed-signal test. For periodic signal, the sampling technique is defined by the following coherent equation:

$$F_t/F_s = M/N, \quad (2)$$

where F_t is the repetitive rate of a periodic signal, F_s is the sampling rate, at which the signal is being sampled, and M and N are integers and they must be prime to each other. To prove that a periodic signal can be completely sampled at a coherent sampling rate, let's denote $\Phi(t)$ as a periodic function with a repeating frequency F_t . The sampling rate F_s is given by the coherent equation (1):

$$\Phi(t) \Rightarrow \Phi[F_t t] = \Phi[(F_t / F_s) \cdot n] = \Phi[(M / N) \cdot n], \quad (3)$$

where $n=0..N-1$.

$$\begin{aligned} &\because \Phi(t) \text{ is a periodic function at } F_t \\ &\therefore \Phi[(M / N) \cdot n] = \Phi[(nM \% N) / N] \\ &\because 0 \leq (nM \% N) \leq N-1, \text{ and } n = 0, (nM \% N) = 0 \\ &\because n_1 \neq n_2, (n_1M \% N) \neq (n_2M \% N) \\ &\therefore (nM \% N) \subset (0, 1, 2, 3, \dots, N-3, N-2, N-1) \end{aligned} \quad (4)$$

Therefore each sample is unique and the collection of the samples represents the complete sample set of a single functional repeat (Fig. 3). One only needs to do a re-order of the samples to recover the waveform. This procedure is so-called reshuffle.

In mixed-signal test, the sampling unit to capture the analog waveform is an ADC or analog-to-digital converter, often called sampler or digitizer and Fig. 3 shows the principle of sampling an analog waveform in mixed-signal test.

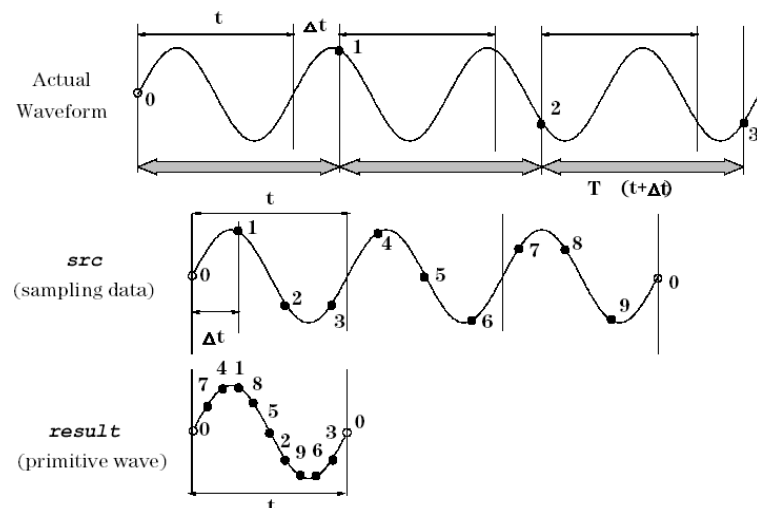


Fig. 3: Coherent sampling technique

The coherent technique is one of the most important techniques to quickly and accurately capture a well-defined periodic signal in the mixed-signal test world.

Now let's look how this concept is applied to a digital channel as 1-bit digitizer in Verigy 93000 SOC system. Because V93000 SOC system is a coherent system in nature, the coherent sampling technique in mixed-signal can also be applied to it. Here instead of moving both timing and level, only level threshold needs to be shifted, this could reduce the test time while achieve a much fine timing resolution comparing to the conventional way.

By adding all the captured digital data at each sampling point up as shown in Fig. 4, an analog waveform can then be digitized. In this way, any PinScale channel can be used like a digitizer or sampler with a bandwidth of the digital channel to perform a mixed-signal type test without adding additional analog instruments. This can be very useful, for example, for high speed device testing in device characterization where signal integrity is of highest priority since it uses exactly the same digital pin without adding any relays to its signal path.

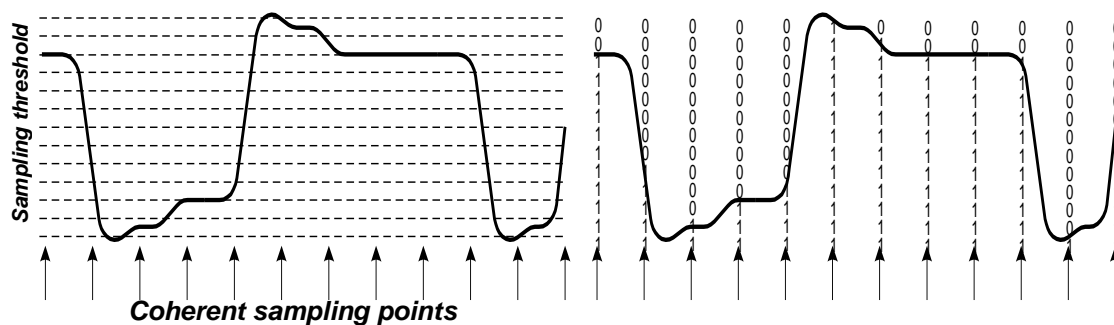


Fig. 4: Waveform reconstruction using coherent technique on digital channel.

The following figures are example waveform plots using coherent sampling technique on digital channels.

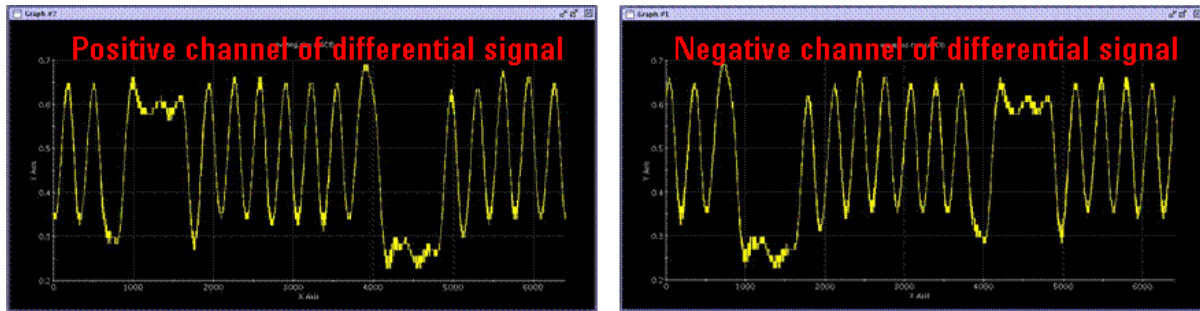


Fig. 5: Waveform captured for a differential signal in the coherent technique

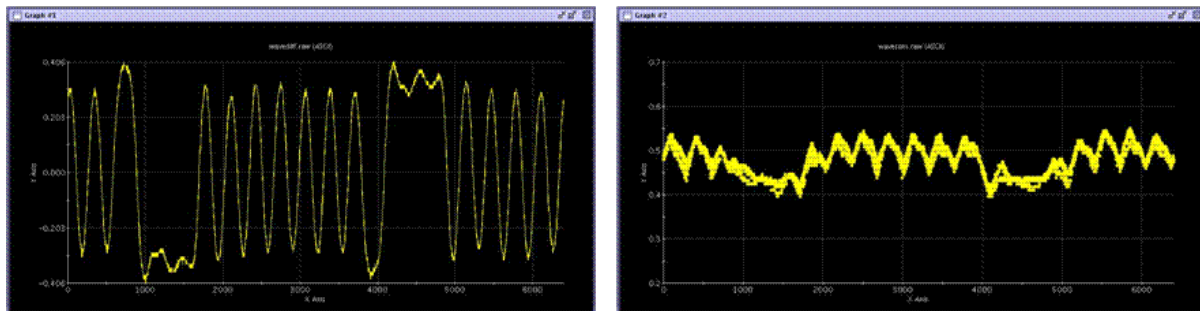


Fig. 6: Differential and common-mode waveform

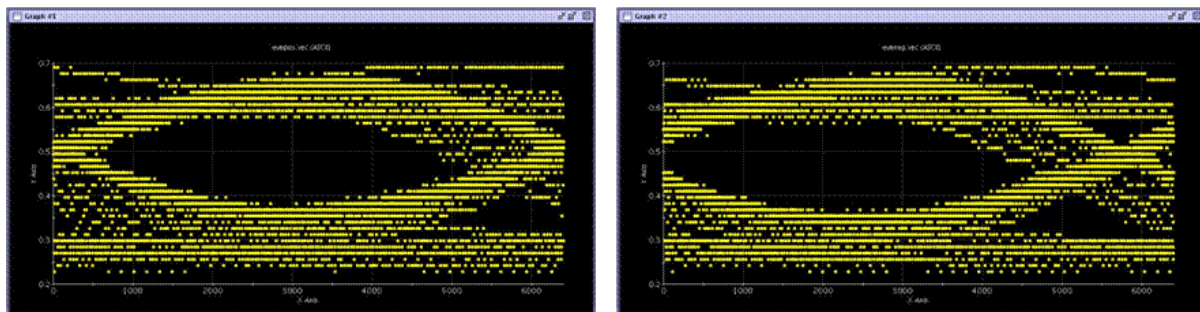


Fig. 7: Eye diagrams of positive and negative channel

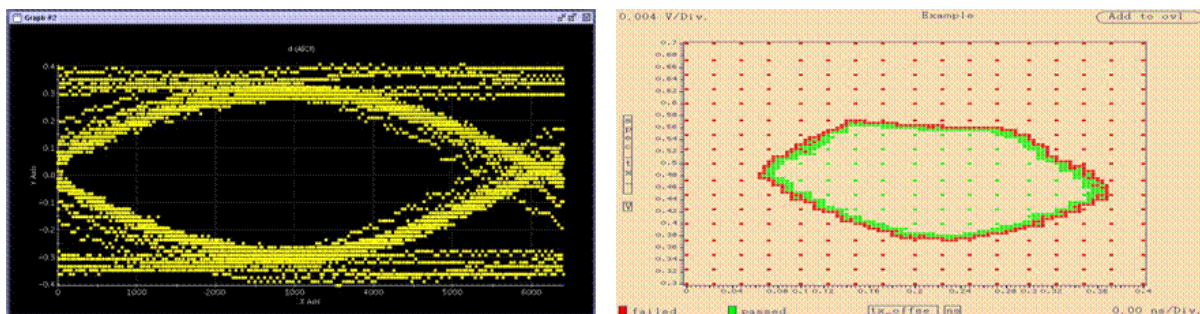


Fig. 8: Eye diagram of differential mode and eye shmoo plot with shmoo tool

Fig. 7 and Fig. 8 show the single-ended and differential data eye. And right plot of Fig. 8 shows the shmoo plot of the single-ended data eye of the same data signal. Clearly they correlate very well with Fig. 7. There is an API developed for digitizing the high speed data signal, calculating eye height and eye width, and graphically displaying data eye plot. There is also a calculator [right in Fig. 9] to calculate M and N value when setting timing equation up in V93000 timing system. The developed API and the tool help to get the M and N numbers quick and correct, and improve the debugging process a lot when digitizing an high-speed waveform using coherent sampling technique by shmoo-plotting each data eye

alongside the execution when it is needed. In this way, design validation engineers can quickly understand how the device performs and where they need to focus on.

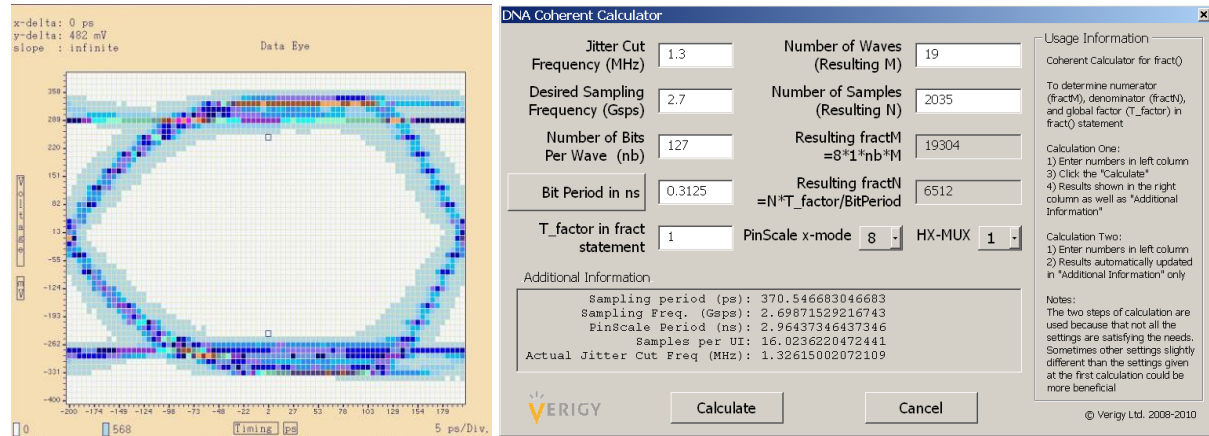


Fig. 9: Eye diagram and coherent calculator using API from DNA tool

Detailed information can be found in the DNA tool programming doc [1].

5 Protocol Aware Solution to address RF test needs

In this section, we will describe how our protocol aware solution fits together with RF device testing in a coherent way. As shown in Fig. 1, an SmartPhone needs an RF component to transmit and receive RF signal from its antenna. The RF signal strength varies from location to location and from time to time. For RF device characterization, often times IC vendors will need to sweep many different settings to evaluate how well the device can tolerate these fluctuation and these settings are the combinations of different parameters. Coherence in this case is not only about the generation of various desired frequencies but also how the various components fit together so that they can play in concert.

An RF chip for the SmartPhone supports different standards such as GSM, EDGE, WCDMA, LTE5 and LTE20 and so on. Normally there are different frequencies transmitting and receiving RF signals called frequency bands or simply bands. Unlike broad-band high speed digital signal, RF signals are narrow banded with several tens of MHz at maximum. Signal transmission path or often called port is specially designed and carefully trimmed or tuned like one listens to a radio device by turning the channel selection knob to select the right channel and to tune the channel right. Only difference though is for SmartPhone that each of these channels uses a different physical path.

Unlike digital test where often times it is to check flip-flops of the device circuits through method of SCAN and so on, RF test is done with register settings through low-speed digital interface such as SPI, JTAG, MDIO etc. There can be overwhelming number of device settings to characterize an RF chip before the device goes for production. Let's assume in average there are 8 different bands for each of these standards. To characterize the device, each band may need to sweep a frequency range, say, 200MHz at 200kHz step which results in 1000 steps. Each of these steps will also need to sweep gain of 50 steps. Ignoring all other settings, we will have total of 5 standards x 1000 frequency steps x 50 gains x 8 ports = 2 million settings. One setting can involve hundreds registers. It is impractical to have all 2 millions settings stored as patterns downloaded into tester hardware for the test. Even worse it becomes a nightmare if for one reason or the other multiple script revisions are generated and need to be verified.

The solution is the Protocol Aware Software solution on V93000 [2]. Many excellent papers on the use of this solution have been published such as [3,4]. With the V93000 Protocol Aware solution, register writes/reads can be done through low-speed digital interface. An example is Serial Peripheral Interface or SPI. It consists of 4 pins: clock pin (CLK), data input pin (MISO), data out pin (MOSI), and chip select (CS) (Fig. 10). V93000 protocol aware software can support this interface with the following setup steps:

- 1) Protocol Definition: defining transactions of the protocol, read and write for each pin with correspondent parameters,
- 2) Use the defined protocol in the V93000 pin configuration to get device pin assigned to correspondent protocol pin,
- 3) Use the protocol software to auto generate protocol wrapper c++ files,
- 4) Use the protocol wrapper in the TestMethod to create on-the-fly protocol pattern to be executed.

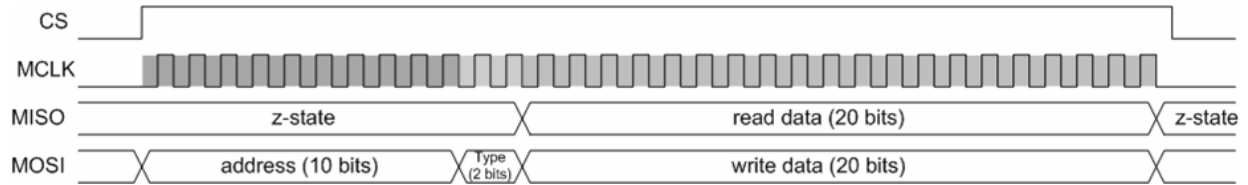


Fig. 10: SPI interface timing diagram

Fig. 11 and Fig. 12 are the representations of same SPI read and write transactions using V93000 protocol software.

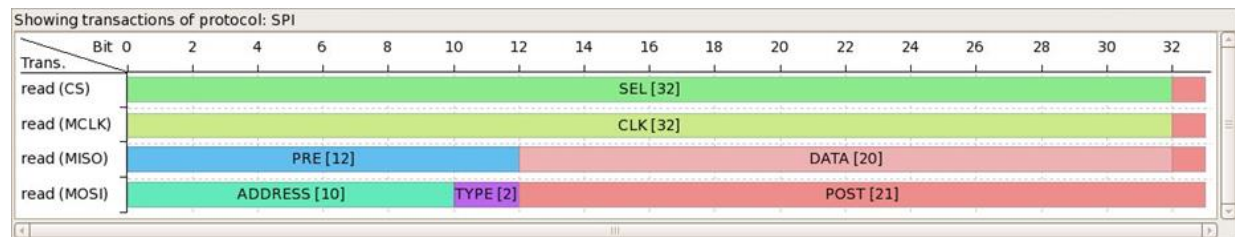


Fig. 11: V93000 Read Transaction Definition

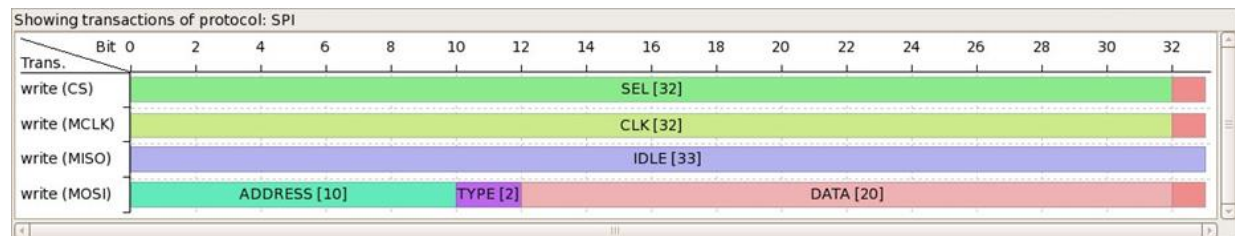


Fig. 12: V93000 Write Transaction Definition

By using protocol software, protocol wrapper can be generated and then can be used in the user TestMethod code. One of such wrapper is displayed in Fig. 13

```

323 TRANSACTION& SPI_WRAPPER_SYNC_TEST::write(UINT64 address, const STRING& type, UINT64 data)
324 {
325     if(pCurrentSequence == NULL)
326     {
327         pCurrentSequence = &(this->add());
328     }
329     TRANSACTION& ta = pCurrentSequence->add("write");
330
331     ta.set("address", address);
332     ta.set("type", type);
333     ta.set("data", data);
334     return ta;
335 }

```

Fig. 13: SPI write wrapper

In addition, wavetable needs to be setup accordingly. Here protocol aware software requires the use of so-called STATEMAP. This is to translate the received protocol bit data to the correspondent device cycle. For details, user can refer to TDC [2] about protocol definition on wavetable.

Now user can use protocol software to create protocol pattern on the fly without a need to preload any vector pattern associated with, e.g. the SPI interface for register settings.

Besides SPI protocol interface, for RF testing, RF subsystem and mixed-signal instruments are required. The RF subsystems are RF front-end and RF source. The mixed-signal instruments are digitizer and AWG. These instruments need to be triggered precisely in association with the setup of the device through device control interface such as SPI interface. Here an RF TRIG protocol is defined. On top of that, there are proprietary digital interface to have RF signal transmitted to and received from base-band processor. Let's call this as DigRF port.

There is a library called SOCLib based on the Concurrent Framework Library developed by R&D to perform this task. At this point of time, we have defined all necessary protocols a) to setup the device (SPI protocol), b) to transmit RF modulation bit stream in digital format (DigRF), and c) to trigger PortScale RF subsystem in the precise timing schedule. The device setup and test procedure are shown in Fig. 16. And this procedure can be repeated for shmooring various register settings.

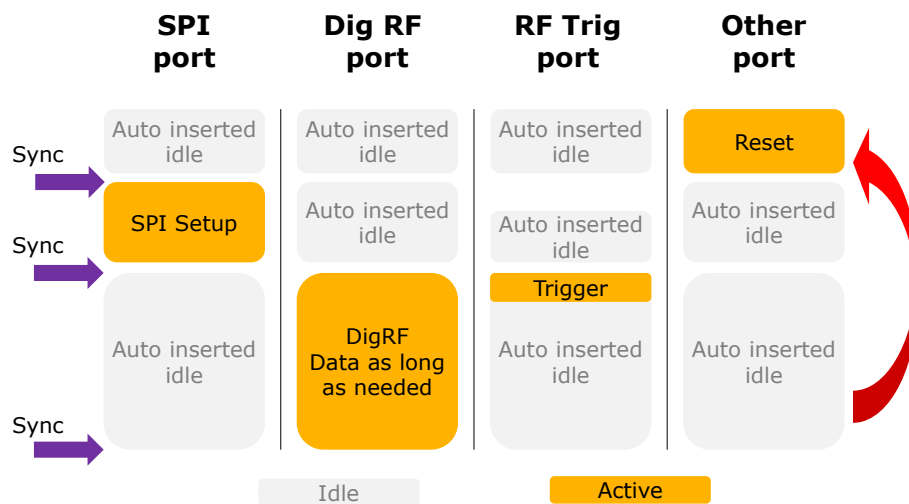


Fig. 14: Timing Schedule of Each Protocol Ports

In order to do this, the library provides an API and this API can take all kind of different protocols either library pre-defined or user-defined protocols and generate needed data for tester channels to perform the test. There are special key words to automatically perform multiple frequencies test or even loop the tests.

It seems that now everything is perfect as described in the papers mentioned above. What's more?

Often times, designers or test engineers change the way how they want to test the chip and generate different revisions of setup files called scripts, and want to give a try. One way to do so is to put these scripts under different folders in the file system in a structured way as visualized in Fig. 15.

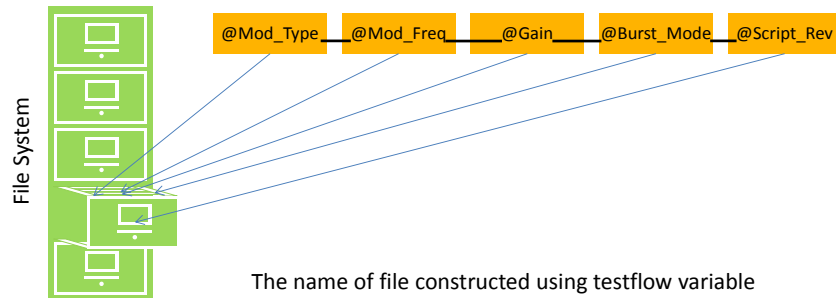


Fig. 15: Constructing file name based on the testflow variables

Testflow variables can be realized using “@” sign. An example is shown in Fig. 16. Thus any revision of scripts can be instantly accessed and the needed settings for a chip can be generated on the fly and online on the tester. Verification of a chip performance can be done smoothly and quickly without a need to re-load millions of pre-converted vector patterns onto the tester and even to re-start the software.

Test Method	SPI_ICELINK\tml.Digital_SPI_ICELINK_Test
Parameters	0, 0 -- Production, EXECUTE, ta_data_@ {DirRev}\RX/WCDMA\ta_@ {ChipRev}\RXp_WCDMA_14
Debug	0
DevelopMode	0 -- Production
ConcurrentMode	EXECUTE
filename	ta_data_@ {DirRev}\RX/WCDMA\ta_@ {ChipRev}\RXp_WCDMA_1476P0_50_@ {RX_ScriptRev}
delay between script	300
Unlink Protocol Patter	YES
Bypass this test	NO

Fig. 16: Select Correspondent Register Settings Using Testflow Variables

Now all pieces of components are working seamlessly and coherently together: the hardware and software to make device characterization and bring-up efficiently and in much quicker speed. This improves the test quality and time to market.

6 Bring Design World and ATE Close Together

As shown in previous two sections, we have described how V93000 can generate test channel frequencies in a precise way after coherent equation to test high speed interfaces and how V93000 can bring different pieces of hardware and software together so that all of them fit well together to improve test quality and reduce the time to market.

Are there still something missing?

One of the biggest hurdles is that there is still a need to generate physical test files or scripts and store them somewhere in file systems, or servers. Often times generating such files takes time and transfers them from one location to other also. Many times different locations simply mean that they are neither in the same country nor in the same time zone. Files are generated and if for some reason there is an issue, people can't verify with each other at the same time. It takes a lot of time to have them all in place. Fig. 17 shows the today's process to verify the test program. It involves many steps.

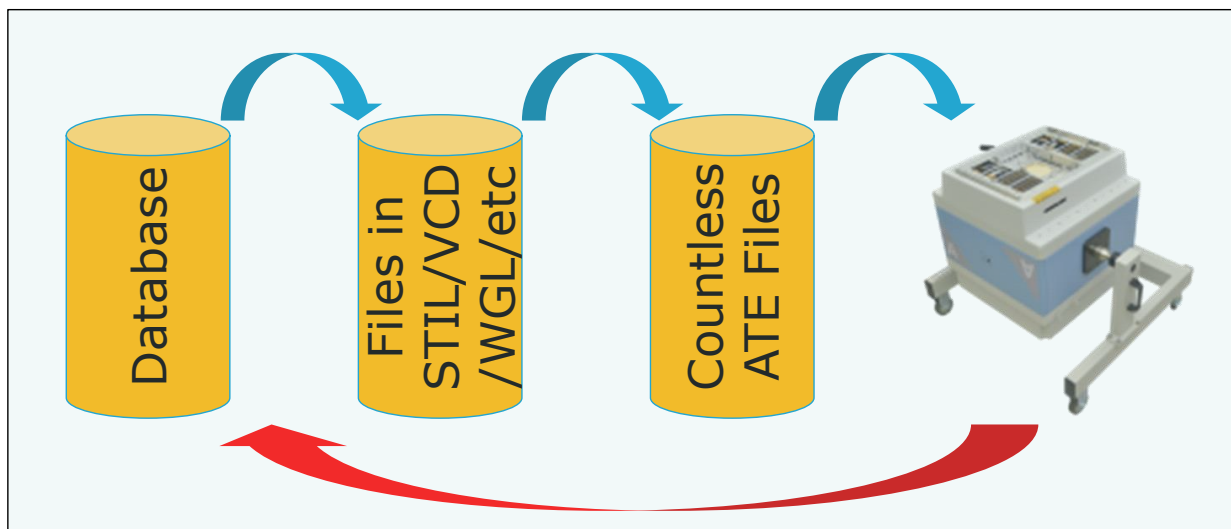


Fig. 17: Today's conventional approach.

Even worse, if there is script issue or pattern issue, test engineers who know how to operate the test system don't have the right skill sets to correct nor it is their task to do so. The designers on the other hand know how to operate the chip but have no clue how to operate an SOC tester. The tedious back and forth between designers and test engineers could take days when not weeks to clarify the mysterious phenomena seen on the tester. This decreases the productivity of engineers involving in the issue and increases the time effort to bring the device to the market.

What's more?

Quite often many of these patterns are just for sole purpose of diagnostics and for design concept verification, they aren't even meant to be seen anywhere in production. Their existence is transient and disappears after a try and probably is never shown up again. All designers want is to literally "shmoo", i.e. to sweep all various settings with various combinations and are trying to see how chip works. In other cases, they want to produce a pre-verified good production test pattern by trying different variety of patterns without even creating them before producing a good pattern which can be eventually used in final production test flow, so that when these verified patterns are transferred to test engineers, they are ready to go in the first place.

However as mentioned, today's way of device verification is always requiring available physical pattern files stored somewhere in the file systems or file servers. In device characterization and verification, much more test patterns are required and this results in massive amount of pattern files to be created and be transferred over the network. It takes additional amount of time to accomplish this work before anything can be tried on the real test system. Often times, designers are just scared and fall back to a "reasonable" amount of test cases which limits the capability of finding potential design flaw, in other words, to the low quality of a test.

What it would be if we could bring designer "directly" to a test system? How about if outputs of an EDA tool which designers are using are re-directed directly to a test system instead writing to a file system, while at the same time, inputs of an EDA tool are made in such a way that it can obtain results directly from a test system instead reading from a file. If this can be made happen, i.e. an interface between designer's tool and test system can be opened, we would have then literally allowed designers to talk to the test system in their own understandable "language" and without even changing their work environment as shown in Fig. 18. Implementation of such link can be realized through computer pipe link technique or TCP-IP protocol over the internet and so on. The list of link technologies out there is not limited.

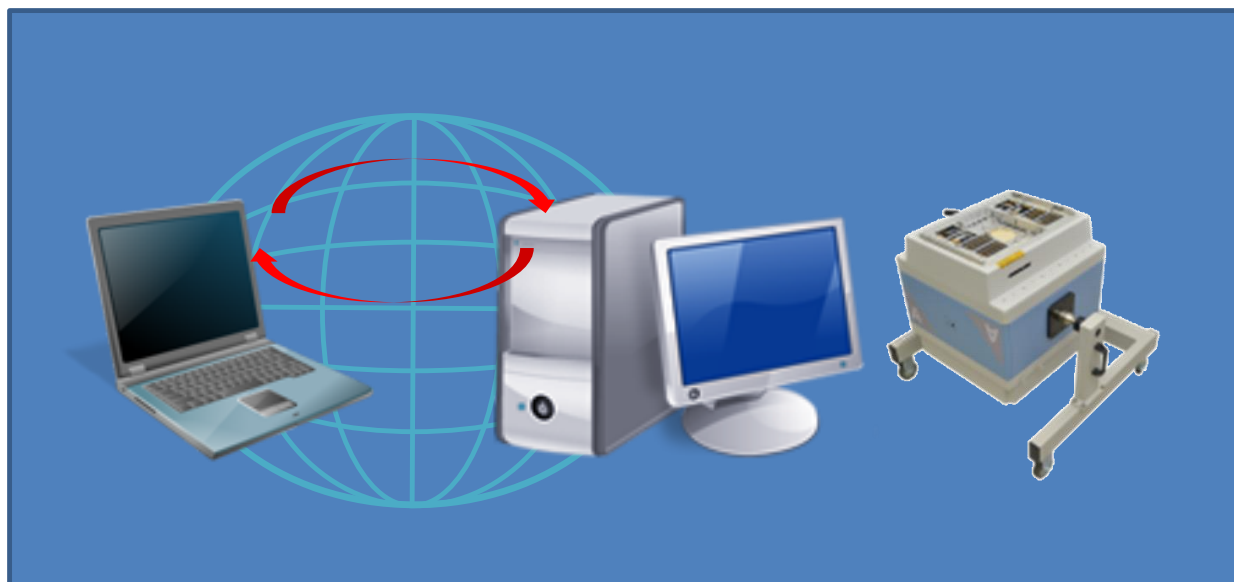


Fig. 18: Link establishment between workstation/PC and tester workstation/PC

A comparison of conventional approach and the process with new interface is illustrated in Fig. 19.

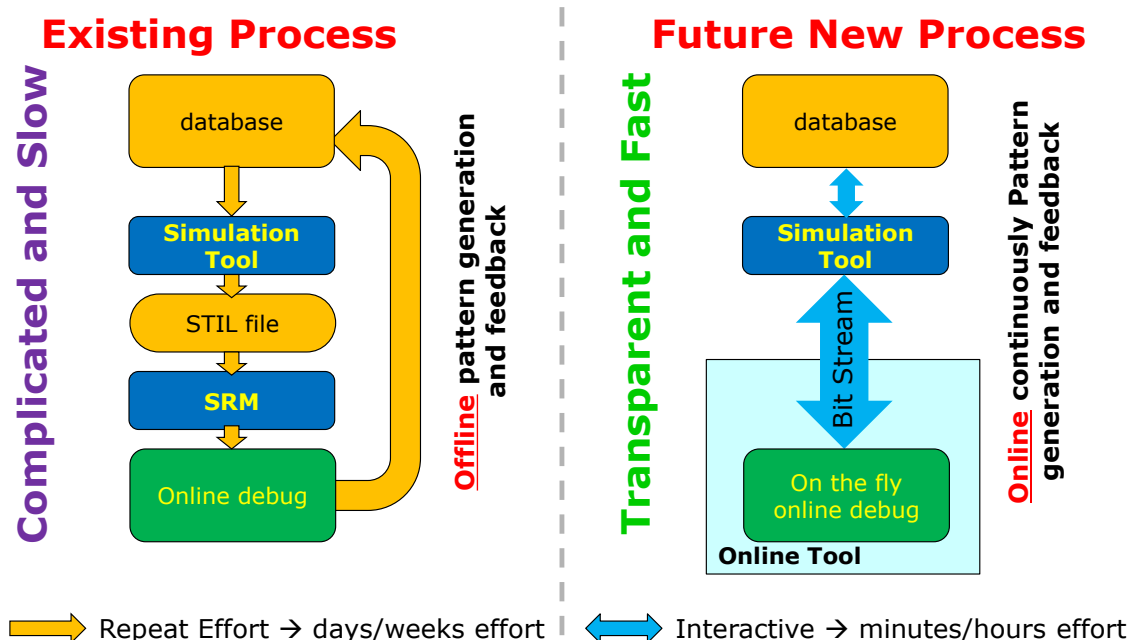


Fig. 19: Design verification process: conventional and new approach

We made this process possible through the translation of pattern on the fly directly to the test system while device under test is connected to the tester and the tester is up and running. By feeding bit stream over computer pipe, fresh patterns are created. We have been working on two different versions of the interfaces to enable the link: like for RF test described before, we created a PA tool using V93000 Protocol Aware Software to interface between the tester and designer's tool; and we created another tool currently called *Smart Design Validation Tool* using V93000 Standard Pattern Translation Software to interface between the tester and designers' tool.

- Same Test Language
- Same Test Program
- Same Test Hardware - LB
- Same Test Software -ST



Engineering ← **Validating** ← **Concepting**

Fig. 20: Coherence from Design to Production

Both tools have been verified and we found pro and cons of both tools. We recommend PA approach when device uses interface such as SPI/JTAG/MDIO for setting registers and doing the testers. We shall recommend direct pattern translation if the pattern transmitted over the link is flat pattern in nature. We have implemented a set of instructions when using this interface. We can work with customers together to

implement the set of instructions to allow the link establishment and communicate to the tester to generate pattern and execute the test, and as well as retrieve the execution results back to the designer's tool.

We have been successfully working out with our customer to drastically change the current work flow from only trying couple of patterns to all kind of combinations of patterns for the test and improve the quality of pattern coming out the designer's tool to test engineering group. It literally eliminates the middle stage of massive amount of time to create, convert, transfer and download these physical patterns. It dramatically reduce the amount of time to debug the pattern because expert is on the line live at the moment when pattern is being generated and being tested immediately afterwards. There is no need for physical patterns for the sole purpose of pattern validation.

7 Summary

In this article, we described three aspects about coherence in testing an SOC devices: wave reconstruction using coherent sampling technique on a pure digital channel, bring different pieces of hardware and software such as register read and write scripts for RF device setup and protocol aware software on V93000 together to allow fast turn-around time during device bring up, device characterization and device high volume production need, and not at least, using computer technique to establish link between device pattern generation tool and SOC tester hardware directly together without a need to intermediary storage of any physical files for concept validation and device characterization.

In order to make them work, all different pieces of components will need to fit well together to reveal the full power of a test system. If one of the pieces doesn't play well in the whole concert, the whole system will not function. If a test system does not allow a precise control of tester channel clock, the coherent sampling technique is only the beauty of a theory. If a system does not provide a full support to allow pattern generation in term of protocol and to allow on-the-fly generation of fresh patterns, such as register read and write task, the RF device test will become a lengthy and error-prone effort to accomplish. If we dare not move step further, we would not explore possibility available in the computer technology out there to enable broad user base in the semiconductor industry.

As of today we have successfully demonstrated and implemented the capability of an SOC system using today's readily available computer technology to allow semiconductor testing fast and transparent.

8 Postscript

The light is composed of millions of different unique frequencies of lights. With the elegant experiment Thomas Young did some two hundred years ago, people realized that light possesses the typical character of wave, thus the term of coherence in optics. The waves moving together produce the day light we are all enjoying. Light hit water droplets in the air atmosphere reflected and refracted produces the beautiful rainbow arc in the sky. This nature phenomena is a result produced by millions of different components working coherently together.

It was a dream of people long before the time of SmartPhone to have a device in the hand to communicate and share information with each other across global such as, e.g. a beautiful rainbow arc seen by someone who can capture it and send instantly to the friends or relatives. It was a vision of people to make such a dreamed device real which contains millions of components instead of millions of waves to coherently fit together in a tiny small device in the hand to produces beautiful way for the people to communicate with each other. The dream leads to a vision and the vision leads to many innovative ideas.

The SmartPhone is such a result of many great ideas. To make the dream real, people, men and women, in semiconductor industry, are working together to allow pieces of the hardware being tested flawlessly. We presented here a system and a concept to link all pieces of components together from design valida-

tion, device bring-up, to high volume production in coherent manner as shown in Fig. 20. We call this as Smart Coherence for SOC Test from design to production on an elegantly designed coherent SOC system: the V93000 SOC System.

9 Reference

1. Jinlei Liu: DNA Tool Programming Reference Rev. 0.952, 2010
2. V93000 Technical Documentation Center, Topic: 125035, 2013
3. Daniel Kather, Jose Moreira, Wen-Jing Song, Roger Nettles, Wei-Min Zhang, Liang Ge: A Unified Approach to Bench/ATE Testing with the V93000 Protocol Editor, VOICE, 2012
4. Alessandro Barboni, Fabio Pizza: Shortening the TTM and increasing the throughput using Protocol Aware, VOICE 2013