Synchronization Error Correction for Asynchronous Channels Data Transmission

Nikolaos Bardis[1,a], *Nikolaos Doukas*[1,b] and *Oleksandr P .Markovskyi*[2,c]

[1] Department of Mathematics and Engineering Sciences, Informatics LAB, Hellenic Military Academy, Greece
[2] Department of Computer Engineering,National Technical University of Ukraine, (Polytechnic Institute of Kiev), Ukraine

Abstract. Loss of synchronization is a common source of errors in asynchronous data channels. Minute differences in the operating frequencies between the transmitter and the receiver result in data bits being lost or false bits being inserted. This paper presents an innovative technique, especially designed for detecting and correcting errors of this type. Data packets are preprocessed and areas that are susceptible to are determined. Suitable redundancy is introduced in the form of control symbols. On the receiver side, similar calculations take place and decisions are made on the occurrence and positions of the transmission errors due to loss of synchronization, which are hence corrected. The proposed method is computationally simple, since it uses a small number of simple mathematical operations, contrary to existing, general purpose techniques. The transmission overheads it imposes do not vary significantly from existing error control codes. Additionally, the number of errors that may be corrected is not subject to the same limits as existing techniques.

1 Introduction

Information transmission is an enabling technology for a variety of systems, most of which are not classified as purely ICT applications, e.g. vehicles, aircraft, ships, weapons systems, production machinery, home appliances etc. The operation of virtually all hardware equipment is heavily dependent on the existence of reliable transmission channels interconnecting subsystems and capable of achieving high transmission rates. Data transmission infrastructure in this context is commonly based on sequential channels that are embedded in the system. The user may often be oblivious to the existence of such embedded transmission channels.

Commonly, serial asynchronous data transmission is the preferred technological basis for embedded channels of the type described above. The characterizing feature of these channels is the fundamental frequency that is employed for controlling this asynchronous transmission. Cost, space and other constraints impose limitations to the number of interface lines used, as well as minimal interface line changes. Serial interfaces that are widely used in such contexts include the Universal Serial Bus (USB) and FireWire. Within the USB protocol, the Non Return to Zero Invert (NRZI) encoding scheme, which is also described by the term *bit stuffing*.

Serial data transmission has been the object of significant technological evolvements and provides ever increasing transmission rates. Typical examples of increasing rates becoming widely available include SATA interfaces, that has during recent years accelerated by a factor of five and the USB 3.0 connection that has increased the standard transmission rate of its previous version by a factor of ten, achieving transmission at 5Gb/s.

Increased transmission rates imply higher error rate values, since the physical media used for the transmission are not in general susceptible to technological evolution and / or are exposed to several types of interference. Hence, low level data transmission is not a very reliable process within modern information processing hardware. Complex physical processes are simultaneously affecting the physical medium, leading to distortions of the transmitted information. It may be stated that in general, larger transmission rates are associated with the existence of more physical processes and phenomena that distort and impede transmission

Classical error correction techniques [1, 2, 3] are not suitable for the case of transmission errors due to loss of synchronization, since they do not take advantage of the nature of the errors in order to increase efficiency. A commonly used method for the prevention of the appearance of synchronization errors is Bit stuffing [4]. This may be implemented by introducing a zero bit after each sequence of binary digits whose length reaches a critical limit, thus changing the dynamics of the

[a] Nikoloas Bardis: bardis@ieee.org
[b] Nikolaos Doukas: ndoukas@sse.gr
[c] Oleksandr P. Markovskyi: markovskyy@i.ua

communications channel. The consequence of this mechanism is that extra bit sequences are transmitted and the non- synchronized bit sequences are interrupted. As a result, the analysis of the received block, including the detection and the correction of the interrupted bits requires a significant amount of overhead computation time.

Specialized methods for the correction of single and double synchronization errors are primarily founded on the principal and the arithmetic of weighted checksums.

More specifically for the correction of single synchronization errors, the two component control code [5, 6] algorithm is used. The first component is computed as the sum modulo 2 of all the bits of the information block. The difference in the first component bit sent by the transmitter and the one calculated by the receiver is used to characterize the type of the error, either the reception of a single extra bit or the loss of a single bit. The second component is calculated as the arithmetic sum of the sequence numbers of all the positions on the original block where unit bits appear in the original block. Using the difference between the received and the calculated second part of the control block, the location of the positions of the erroneous synchronized bits is performed.

An alternative existing method [7, 8, 9] is capable of correcting double synchronization errors and uses a two component correction code. With the bit values of the first component, the fragments of the transmitted packet that contain errors. The second component bit determines the type of each synchronization error, namely the increase or decrease of the received data by a bit with value 1.

The disadvantage of these methods is that the number of the errors that may be corrected is limited. At the same time, there exist multiple applications for which the transmitter and the receiver operate at significantly different temperature conditions, hence presenting increased probability of error appearance, with multiplicities greater than 2.

Therefore, existing methods do not provide in a large number of instances, effective solutions for the problem of detection and correction of transmission errors due to the loss of synchronization.

The purpose of this research is to develop an effective method for the correction of all errors due to loss of synchronization in the transmission rates of data between computer systems that use asynchronous data.

2 Development of the method

For the purpose of attaining this aim, a simple and efficient method for correcting all errors due to loss of synchronization that occur during the sequential transmission of an n – bit packet in an asynchronous channel.

Let $B_R = \{b_1, b_2,\ldots,b_n\}$, $\forall k \in \{1..n\}$: $b_k \in \{1, 0\}$be the data that are taken into account in the analysis of the control sequence at the transmitter and the receiver. The method enables the detection of errors due to synchronization, determines their type and executes the required corrections, independent of the number of errors in the packet.

Modern sequential data exchange protocols are synchronized via the sign bits – zero or one. The transmission of bits of the opposite sign do not incur a sign change on the line, which implies that their transition is not synchronized. For example, USB channels are not synchronized during the transmission of unit bits. The total length of the sequence of units is determined at the receiver by the duration of the constant voltage on the line. Эта длительность на приемнике и передатчике измеряется схемой таймера The timers at the transmitter and the receiver may differ. Измеренная длительность постоянного потенциала на приемнике и передатчике может отличаться. This problem becomes more pronounced when the transmitter and receiver timers operate in different temperatures. The result of this discrepancy is that the number of the subsequent orders of units at the receiver may differ from the number of units at the transmitter. The synchronization errors hence appear.

In practice, the length difference of the transmitted and received packets, that contain a sequence of unit bits, does not contain more units. The probability of error appearance increases with increasing lengths of unit bit sequences. If the length l tends to a critical threshold h there is a realistic risk of synchronization errors appearing. For this reason, it becomes more difficult to use specialized devices. For a USB port and approximately equal transmitter and receiver temperatures, the critical threshold is six units: h=6 [8].

For the transmission of an n-bit data packet, the appearance of synchronization errors is possible exclusively for packets that contain the sequence of unity bits and their length is greater than or equal to the critical length h.

For this reason, the developed approach proposes the determination of a fragments transmitted within the block of data BS that contain sequential unit bits and have a length greater than or equal to $h - 1$.

The determined fragments may be assigned names E_{1S}, E_{2S}, …, E_{mS}, where m is the number of n – bit fragments within the block BS. The lengths of the fragments are correspondingly assigned to the variables: $l_{1S}, l_{2S}, \ldots, l_{mS}, \forall i = 1, \ldots,m; l_{iS} \geq h - 1$.

On the receiver side, it is proposed that a control sequence S is calculated that consists of two bit symbols $c_{1S}, c_{2S}, \ldots,c_{mS}$, where $c_{iS} \in \{0,1,2,3\}$, $i \in \{1,..,m\}$ and each one of those symbols is calculated as the remainder of the division of the length of the i^{th} fragment by four.

$$c_{iS} = l_{iS} \bmod 4 \quad (1)$$

The control sequence S is transmitted along with the information block BS.

The received information block (BR) is analyzed correspondingly to the structure of the transmitted block: the fragments E_{1R}, E_{2R}, ... E_{mR} are parsed that should be populated by sequences of unit bits whose length is not shorter that h –1. The lengths of these fragments are

assigned to variables l_{1R}, l_{2R}, …, l_{mR}, $\forall i = 1, \ldots, m$; $l_{iR} \geq h-1$.

On the receiver side, using the received block BR, it is proposed that the control sequence R is calculated, consisting of the two bit symbols c_{iR}. In this case $i = 1 \ldots m$ and each c_{iR} is calculated as the remainder of the division of the length l_{iR} of the i^{th} fragment E_{iR} by four:

$$c_{iR} = l_{iR} \bmod 4 \quad (1)$$

The detection of the errors in the received information block, the determination of their type, as well as the correction are implemented on the receiver side. In the i^{th} received fragment, E_{iR}, it is possible to observe the following two types of errors due to timing: the appearance of additional unity bits and the subsequent increase in the length of the received fragment compared to the transmitted fragment, as well as the loss of a unity bit which means that the received fragment is one bit shorter than the transmitted one, i.e. $l_{iR} = l_{iR} - 1$.

The solution of the exercise of the detection of synchronization errors is proposed to be performed via the comparison and analysis of two control sequences: the one received from the transmitter and the one calculated at the receiver. If all the corresponding symbols are equal, then the solution implies is that there were no transmission errors: $\forall i \in \{1,2,..,m\}$: $c_{iS} = c_{iR}$. In other occasions, if for every i symbol pairs $c_{iS} \neq c_{iR}$, the result is interpreted as the existence of synchronization errors for the i^{th} fragment. For the determination of the types of errors, the following procedure is proposed: the hypothesis is tested that the length of the sequence of units in the i^{th} fragment has been increased on the receiver side. For this reason a unit is added in the i^{th} control symbol c_{iS} of the sequence of the transmitter and the calculation of the remainder of the division of the augmented symbol modulo 4. The result is compared with the corresponding symbols of the control sequence of the receiver c_{iR}. In case these are equal i.e. $(c_{iS} + 1) \bmod 4 = c_{iR}$, the above hypothesis is assumed to be true and for this reason the decision is made to correct the i^{th} fragment of the received packet by eliminating the excess unit bit. If on the other hand, $(c_{iS} + 1) \bmod 4 \neq c_{iR}$ and $c_{iS} \neq c_{iR}$ the decision is made that a different type of synchronization transmission error occurred and the number of ones of the i^{th} fragment was reduced on the receiver side: for the correction of this error a unit bit is added to the i^{th} fragment of the packet on the receiver side.

Formally, the proposed method for the correction of transmission synchronization errors may be summarized as the following sequence of steps:

1. In the transmitted packet BS, the fragments E_{1S}, E_{2S}, …, E_{mS} are determined in which the number of sequential unit bits is greater than $h-2$.
2. Determination of the length l_{iS} of the i^{th} fragment E_{iS}, $i \in (1 \ldots m)$ for the transmitted packet.
3. The two bit sequence control symbols $c_{1S}, c_{2S}, \ldots, c_{mS}$ are calculated for the sequence S on the sender side, by obtaining the remainder of the division of the length l_{iS} by four for every fragment i.
4. The packet BS is transmitted together with the control sequence symbols of S $c_{1S}, c_{2S}, \ldots, c_{mS}$.
5. In the received packet BR, correspondingly to step 1 the fragments E_{1R}, E_{2R}, …, E_{mR} are determined that consist of sequences of unit bits the length of which is larger than $h-2$.
6. Correspondingly to step 2, the number of unit bits in each fragment is determined: E_{1R}, E_{2R}, …, E_{mR}. These are essentially equal to $l_{1R}, l_{2R}, \ldots, l_{mR}$, $\forall i = 1, \ldots, m$; $l_{iR} \geq h-1$.
7. The control sequence R of the receiver consists of the two bit symbols c_{1R}, c_{2R}, …, c_{mR}. Each symbol c_{iR} is calculated as the remainder of the division of the length l_{iR} by four for every fragment i.
8. The counter j used for processing symbols c_{1R}, $c_{2R}, \ldots, c_{mR}$ is initialized to 1.
9. If $c_{jS} = c_{jR}$ then proceed to step 12
10. If $(c_{jS} + 1) \bmod 4 = c_{jR}$, then the fragment j was transmitted with a synchronization error: at the receiver the number of unit bits of the j^{th} fragment was increased by 1. Consequently, the error is corrected by decreasing the unit bits of the j^{th} fragment E_{jR} by 1. Proceed to step 12.
11. The fragment E_{jR} suffered the opposite synchronization error: during transmission, the number of unit bits of the j^{th} fragment was reduced by 1. Hence a unit bit is added to the j^{th} fragment E_{jR}.
12. If $j < m$, increment the counter j and return to step 9.

End of procedure.

The proposed method may be illustrated via the following example:

Let the critical limit h for the existence of risk for the occurrence of synchronization errors be defined as $h=6$. Furthermore, assume that the transmitted packet size is 64 bits, BS = {00**11 1111 11**00 0010 0**111 1111 111**0 0100 10**11 1111** 000**1 1111 11**01 0101 00**11 1110**}.

According to the proposed method, the transmitted packet BS is processed to determine the unit bit sequences the length of which is not shorter than $h - 1 = 5$. The packet BS contains 5 ($m = 5$) such sequences E_{1S}, E_{2S}, …, E_{5S} that are marked with boldface. The numbers of units in each fragment are calculated as: $l_{1S} = 8$, $l_{2S} = 10$, $l_{3S} = 6$, $l_{4S} = 7$, $l_{5S} = 5$. The symbols c_{1S}, c_{2S} …, c_{5S} of the control sequence S are calculated as the as the remainder of the division of l_{iS} by 4 for each fragment i: $c_{1S} = l_{1S} \bmod 4 = 8 \bmod 4 = 0$, $c_{2S} = l_{2S} \bmod 4 = 10 \bmod 4 = 2$, $c_{3S} = l_{3S} \bmod 4 = 6 \bmod 4 = 2$, $c_{4S} = l_{4S} \bmod 4 = 7 \bmod 4 = 3$, $c_{5S} = l_{5S} \bmod 4 = 5 \bmod 4 = 1$. Consequently, the control sequence S consists of the following five symbols: $S = \{0, 2, 2, 3, 1\}$.

Consider for example the incidence where the receiver receives the following erroneous data packet: BR = {00**11 1111 111**0 0001 00**11 1111 111**0 0100 10**11 1111 1**000 **1111 11**01 0101 00**11 1110**}. In the received packet BR, the fragments are determined as sequences of unit bits of length not less than $h-2$. The received packet BR

contains 5 such sequences (m = 5) to be considered, E_{1R}, E_{2R} …, E_{5R}, that are marked in boldface. The length of these fragments is correspondingly equal to l_{1R}=9, l_{2R}=9, l_{3R}=7, l_{4R}=6, l_{5R}=5. The control sequence R, consisting of the symbols c_{1R}, c_{2R}, …, c_{5R}, is formulated by calculating the remainder of the division of l_{iR} by four for each fragment i: c_{1R}=l_{1R} mod 4=9 mod 4=1, c_{2R}=l_{2R} mod 4=9 mod 4=1, c_{3R}=l_{3R} mod 4=7 mod 4=3, c_{4R}=l_{4R} mod 4=6 mod 4=2, c_{5R}=l_{5R} mod 4=5 mod 4=1. The obtained control sequence R is hence calculated as containing the following symbols: R = {1, 1, 3, 2, 1}.
The counter is initialized: j = 1.
According to Step 9, the relationship between corresponding symbols of the two control sequences S and R: in the case of c_{1S} and c_{1R}, since c_{1S}≠c_{1R} (0≠1), the 1st fragment of the received packet BR contains a synchronization error. The determination of the type of the error is performed according to Step 10, by testing the hypothesis (c_{1S}+1) mod 4=c_{1R}. For the first pair of control symbols the hypothesis is verified since (0+1) mod 4=1 (1=1). It is hence determined that the received packet contains one extra unit bit in the fragment E_{1R}. The correction of this error is performed by removing the surplus bit from the fragment E_{1R}.
Since the value of the counter is smaller than the number of fragments m, the counter is incremented, i.e. j = 2.
The relationship between the second pair of symbols of the control sequences S and R, c_{2S} and c_{2R} is investigated: c_{2S} ≠ c_{2R} (2≠1) and a synchronization error is determined to exist in the second fragment of the received packet BR. The condition defined in Step 10 is tested (c_{2S} + 1) mod 4 = c_{2R}, that in this case is found not to be satisfied: (2 + 1) mod 4≠1 (3≠1). According to Step 11, it is concluded that the fragment E_{2R} has been received with a synchronization error: the number of unit bits in the fragment has been reduced by one on the receiver side. The elimination of the error is achieved by inserting a unit bit in the fragment E_{2R}.

S $j < m$ the counter is incremented to j = 3.

According to Step 9 the relationship between the third pair of control symbols c_{3S} and c_{3R} of the control sequences S and R is investigated. Since these are not equal, c_{3S}≠c_{3R} (2≠3), the decision is made that the received fragment E_{3R} contains a synchronization error. For the determination of the type of the error the condition (c_{3S} + 1) mod 4=c_{3R} is investigated, that is in this occasion verified: (2 + 1) mod 4=3 (3=3). According to Step 10 the error is determined to be manifested as an extra unit bit in the received fragment E_{3R}. A correction is performed in the 3rd fragment of the received packet BR, by removing one non – synchronized bit.

The value of the counter j is less than m: 3 < 5; hence the counter is incremented (j = 4).
The relationship between the fourth (j = 4) pairs of control symbols c_{4S} and c_{4R} of the control sequences S and R is investigated: since c_{4S}≠c_{4R} (3≠2), the 4th fragment is detected to contain a synchronization error and requires correction. According to Step 10, the condition (c_{4S}+1) mod 4=c_{4R} is not satisfied since (3+1) mod 4≠2 (0≠2). This error is classified as a synchronization error whereby the number of unit bits in the 4th fragment has been reduced by one during transmission. For the correction of the detected error, a unit bit is inserted into the fragment E_{4R} of the received packet.
According to Step 12, the value of the counter is incremented: j = 5.
According to Step 9: since c_{5S} = c_{5R} , it is hence verified that the fragment E_{5R} was transmitted without any synchronization errors. The calculation jumps to Step 12.
As the value of the counter j is equal to the number of fragments m, the error correction process terminates.
In summary, for the presented example it was detected that the received packet BR contained four synchronization errors in five fragments:

- In fragment E_{1R}, an additional unit bit was created
- In fragment E_{2R}, a unit bit was deleted,
- In fragment E_{3R}, an additional unit bit was created
- In fragment E_{4R}, a unit bit was deleted
- In fragment E_{5R}, there were no errors appearing

The information packet at the receiver side will hence be reconstructed as: BR = {0011 1111 1100 0010 0111 1111 1110 0100 1011 1111 0001 1111 1101 0101 0011 1110}, which is an exact match of the transmitted packet BS.

3 Efficiency analysis

The principle advantage of the proposed synchronization error correction method, compared to existing techniques is the lack of limitation in the number of errors that may be corrected. All current techniques for error correction have been designed so as to correct up to two transmission errors. The proposed method does not require any assumptions on the numbers of errors that may have occurred during data transmission. This fact is particularly important for real time system and procedure control applications, components of which operate in different temperatures. More specifically, relevant applications include aircraft control, weapons systems and C4ISR. In such applications, the number of digital sensors involved may reach 250,000. Systems of this scale may be described as complex digital systems for the control of devices and technological processes that are associated with extreme temperature conditions.

The number k of control symbols that are transmitted together with the information packet, is a random variable that depends on the length n.

Assuming that the values of zero and one appear with equal probability for any bit in the information packet, it may be shown that the average number of fragments that consist of exactly j unit bits is equal to $n/2^{j+2}$. Consequently, the mean number k of the fragments of the information packet that consist of a sequence of unit bits of length that is not less than h – 2, is estimated as the sum:

$$k \approx \sum_{j=h-1}^{\log_2 n+5} \frac{n}{2^{j+2}} = \frac{n}{2^h} \quad (3)$$

Hence, taking into account that for each sequence at least $h - 2$ unit bits convey the digital symbol, the mean number of redundant information bits is $n / 2^{h-1}$ or the percentage of the redundancy $2^{-(h-1)} \cdot 100\%$. For example, if a common value of $h = 6$, for the controlled transmission of a 256 byte long packet ($n = 2048$), according to (3), the mean number k of the length of the sequence of units of length of no less than $h - 1 = 5$ is $k = 32$. Consequently, the proposed method requires the transmission of an additional number of $k = 32$ double bit symbols or 64 bits that correspond to an overhead of $2^{-(h-1)} \cdot 100\% = 3.125\%$.

Existing systems that are capable of only correcting double synchronization errors, use for similar circumstances $2 \cdot \log_2 k = 64$ control bits and therefore impose a level of redundancy that is similar to the proposed method. Using bit stuffing, an additional bit is required for each sequence of length greater than $h - 2$ unit bits, i.e. $k/2 = 16$ extra bits. The proposed method is therefore shown to possess high applicability for the correction of transmission synchronization errors and the level of redundancy imposed as overhead is not significantly different in practice, compared to existing methods.

One of the principle advantages of the proposed method, in contrast with existing ones, is the use of simple mathematical operations. In total, the detection of synchronization errors requires on average two comparison operations, one addition plus one more logical operation. Hence the proposed error correction procedure is mathematically simple and requires significantly smaller computational complexity, compared to existing ones. Compared to bit stuffing, the proposed method offers, for medium synchronization error occurrence rates, a significant acceleration of the error control procedure, because of the use of much simpler mathematical operations.

4 Conclusions

Asynchronous serial transmission is a fundamental technology that enables the use of embedded information processing equipment in diverse environments. Transmission errors due to loss of synchronization are a special class of errors, to which serial asynchronous lines are especially susceptible and represent an important impediment to the achievement of increased transmission speeds in such transmission lines. Classical error correction techniques can only correct limited numbers of errors and impose significant computational overheads. A technique especially designed for correcting transmission errors due to loss of synchronization was presented. The technique is based on segmentation of the information packet and the insertion of suitable error control codes. This technique provides capabilities for correction of larger numbers of errors than classical techniques, is computationally more efficient and imposes less transmission overheads than bit stuffing, which is the alternative, specialized technique for correcting transmission errors due to loss of synchronization. The technique was illustrated by means of an example and was shown to be capable of correcting four such errors, of two different types within a 64 bit packet. Simulation and experimental results will be pursued so as to demonstrate the applicability of the proposed method.

References

1. Klove T. Error Detecting Codes: General Theory and Their Application in Feedback Communication Systems / T. Klove, V. Korzhik.- Norwell, MA: Kluwer, 1995. – 433 p.
2. A. J. McAuley, Weighted sum codes for error detection and their comparison with existing codes, IEEE/ACM Trans. On Networking, vol. 2, no. 1, pp. 16–22, Feb. 1994.
3. Zumbragel J. Skachek V., Flanagan M. On the Pseudocodeword Redundancy of Binary Linear Codes J. Zumbragel, V. IEEE Trans. of Information Theory.-2012.-Vol.58.- № 7.- P.4848-4861.
4. R. Prévost, M. Coulon, D. Bonacci, J. LeMaitre, J. P. Millerioux and J. Y. Tourneret, "CRC-assisted error correction in a trellis coded system with bit stuffing," 2011 IEEE Statistical Signal Processing Workshop (SSP), Nice, 2011, pp. 381-384. doi: 10.1109/SSP.2011.5967710
5. Bardis N.G, Markovskyy A.P, "Utilization of Avalanche Transformation for Increasing of Echoplex and Checksum Data Transmission Control Reliability", ISITA 2004 - INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY AND ITS APPLICATIONS, IEEE / SITA, Parma, Italy, October 10-13, ISBN: 4-902087-08-1, 2004
6. Nikolaos G. Bardis, Athanasios Drigas and Oleksandr P. Markovskyi, Performance Increase of Error Control Operation on Data Transmission, 3rd International Conference on New Technologies, Mobility and Security, IEEE, IEEE COMMUNICATIONS SOCIETY, Egypt – Cairo, 20-23 December 2009.
7. Bardis N. G, Doukas, N., Markovskyi O.P, Transmission Error Correction Based on the Weighted Checksum, Proceedings of the APPLIED COMPUTING CONFERENCE 2009 (ACC '09), MATHEMATICAL METHODS AND APPLIED COMPUTING Volume II, Vouliagmeni-Athens, Greece, pp.758-763, September 28-30, 2009.
8. Nikolaos Bardis, Nikolaos Doukas, Olexander Markovskyi, “Efficient Burst Error Correction Method for Application in Low Frequency Channels and Data Storage Units”, DSP/IEEE 2011 17th International Conference on Digital Signal Processing, 6-8 July 2011 Corfu, Greece.

9. Olga Fedorechko, Olexandr Markovskyi, Nikolas Doukas, Nikolas Bardis. Synchronization Error Detection of Data Transmission Errors in Asynchronous Channels. Proceedings of the 18th International Conference on Systems (part of CSCC '14) Santorini Island, Greece July 17 - 21, 2014