GAS Journal of Engineering and Technology (GASJET)

Volume 2, Issue 7, 2025 Journal Homepage: https://gaspublishers.com/gasjet-home/ ISSN: 3048-5800
Email: editor@gaspublishers.com

Synchronous Ethernet-Based Frequency Synchronization in Packet Transport Network for Mobile Backhaul

UnChol Song, JongSam Ri, NamChol Yu*

Kim Chaek University of Technology, Pyongyang, Democratic People’s Republic of Korea

Received: 29.08.2025 | **Accepted:** 05.09.2025 | **Published:** 24.09.2025

***Corresponding Author:** NamChol Yu

DOI: *10.5281/zenodo.17186343*

Abstract **Original Research Article**

A packet transmission network (PTN) is a transmission network based on packet switching technology, when using a packet transmission network for mobile backhaul (MBH), all nodes in the network must have frequency synchronization to achieve base-station synchronization through the network. Synchronous Ethernet (SyncE) is the most efficient frequency synchronization method available in a packet transmission network, where each node in the transmission network regenerates clock information at the physical layer level, filters jitter and offset through the phase-locked loop (PLL), and then delivers frequency by distributing it across the physical layer to the next nodes regardless of the higher layer transmission protocols. In this paper, in order to implement synchronous Ethernet-based frequency synchronization in packet transmission network, the synchronization block of 1G and 10G transmission equipment is composed of synchronous clock generation unit, jitter rejection and frequency multiplication unit, synchronous clock generation unit is implemented on FPGA (Field Programmable Gate Array) chip, jitter rejection and frequency multiplication unit is implemented on Si5328 chip, jitter attenuation precision clock synthesizer. Given that the PLL resource embedded in the FPGA chip is limited and does not satisfy the multiplication ratio required to obtain the synchronous clock from the receive regenerative clock outputted from the 10G physical layer device, the clock synchronized to the receive regenerative clock outputted by the physical layer is generated using only pure logic elements based on the Numerical Controller Oscillator (NCO) principle, so we have reduced PLL’s resource consumptions using this way. The jitter rejection and frequency multiplication unit eliminates the phase jitter from the synchronous clock to the receive regenerative clock detection unit and the synchronous clock generation unit, and sets the parameters of the Si5328 chip necessary to synthesize the transmitting clock of the physical layer device for synchronous Ethernet. In addition, the experimental system was constructed and measured with 1G and 10G transmitting nodes applying the proposed synchronization method. As a result, phase fluctuation was 23ns, which is consistent with the requirement of synchronous Ethernet specified in ITU-T Recommendation G.8262.

Keywords: Packet Transport Network (PTN), SyncE, Mobile Backhaul (MBH), Frequency Synchronization.

1. Introduction

Packet Transport Network (PTN) is a transport network that converts data, voice, video, and other services into packets and then transmits them using MPLS-TP technology [1]. Packet transport networks are a new type of transport network that combines the features of packet networks such as Ethernet and traditional transport networks such as SDH which can operate in asynchronous or synchronous mode from the synchronous side. In a packet transport network, the requirement for synchronization may vary depending on the type of service. In particular, when using packet transmission networks for mobile backhaul (MBH), the synchronization requirements are stringent [4]. Mobile communication systems require frequency and time synchronization for a proper operation. In mobile networks, frequency synchronization is firstly needed to synchronize the wireless carrier between the base stations. If the wireless carriers of the base stations are not synchronized with each other, the bands allocated to the user at the base stations may overlap each other and thus cause inter-channel interference [2]. In mobile networks, frequency synchronization is also required to prevent call failure and voice interruption during handover. In general, long handover times can lead to speech interruptions or call failures. The handover time is much shorter (less than 100 ms) when the current base station and the destination base station are synchronized, which can smooth the handover [3]. The time and frequency synchronization between the base stations can be done by each base station using a GPS receiver (Fig. 1), another method is to use a GPS receiver only at the central node, and the base stations can perform time and frequency synchronization through MBH (Fig.2). In the case of timing and frequency synchronization across the transport network, each node within the transport network must have timing and frequency synchronization functions.

Although GPS can provide high time and frequency accuracy, there are several problems with GPS receivers, such as the fact that the installation of GPS receivers at all base stations requires a lot of cost, and the fact that the base stations are located in the GPS dead zone which affects the whole communication system in case of GPS signal jamming [8]. Therefore, mobile communication systems are currently using the synchronous mode through MBH. It is important to implement frequency synchronization functions for all the transmitting nodes within the MBH. The MBH is a network that transfers user data and control data between the Radio Access Network (RAN) and the Core Network (CN) of the mobile network, which can use a time division multiplexed circuit switched network or an IP-based packet switched network. The frequency synchronization methods used in packet transport networks include GPS clocks, BITS (Building Integration Timing Supply) clocks, E1 clocks and other external synchronization clocks, SyncE (Synchronous Ethernet) and packet-based methods [4–7]. A packet-based approach is to synchronize specific packets by sending their timing information which includes the Network Time Protocol (NTP) and the IEEE 1588 Precision Time Protocol (PTP) [14, 21–24]. The NTP protocol achieves a synchronization in a soft way, which leads to poor synchronization accuracy [24]. The currently used NTP (RFC4330:NTPv4) has a time accuracy of about 10 ms, which is not appropriate for the time accuracy requirements of mobile networks. The IEEE1588 PTP increases the synchronization accuracy by stamping the departure time on packets in hardware instead of overcoming the disadvantage of the NTPs low synchronization accuracy in software [14]. The PTP can achieve high frequency accuracy of less than 50 ppb (parts per billion) from its characteristics [26]. Table 1 lists the time and frequency synchronization methods available for packet transport networks.

In this paper, we describe the challenges of implementing a frequency synchronization system based on SyncE, which is one of the frequency synchronization methods in packet transport networks. The rest of this paper is organized as follows: Section 2 describes the general principles and timing requirements of SyncE. Section 3 presents the challenges of SyncE-based FSS implementations such as timing signal detection, synchronous clock generation, and frequency synthesis. Section 4 builds a test synchronous network and analyzes the performance of the proposed synchronous system. Section 5 concludes this paper.

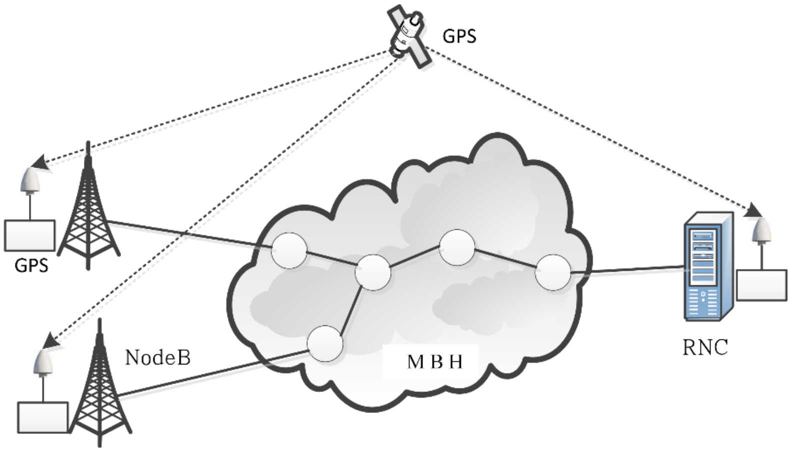


Fig.1. Synchronization mode with GPS.

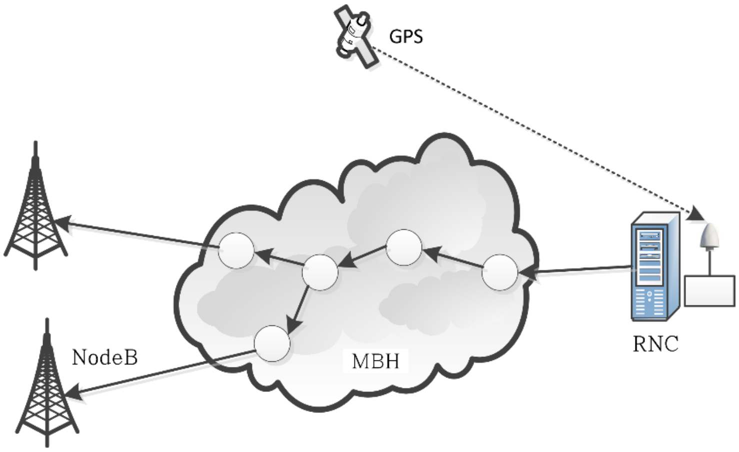


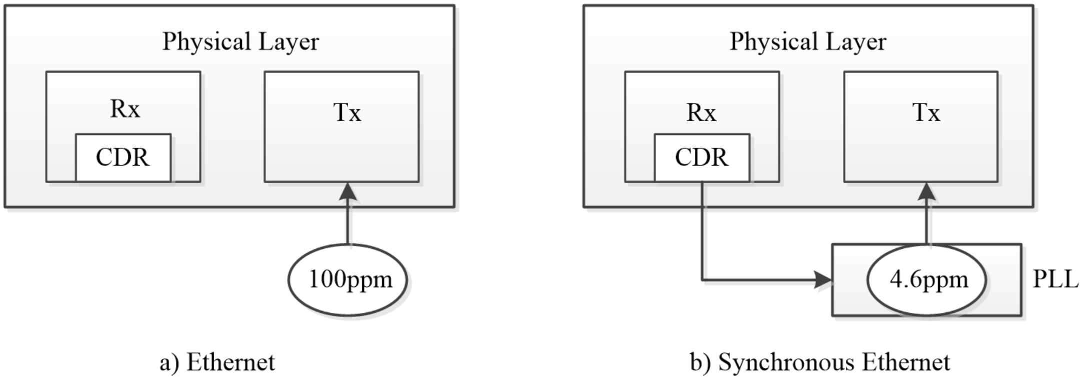
Fig.2. Synchronous mode through MBH

Table 1. Synchronous methods in packet transport networks

Synchronization method	Frequency synchronization	Time synchronization
GPS	possible	possible
BITS	possible	impossible
SyncE	possible	impossible
NTP	impossible	possible
PTP	possible	possible

2. Synchronous Ethernet

Synchronous Ethernet is a proposed technique for frequency synchronization at the physical layer level [7, 9–11]. Fig. 3 shows the principle of SyncE compared with conventional Ethernet.



a) Ethernet b) Synchronous Ethernet

Fig.3. Synchronization flow in Ethernet

In Generic Ethernet, the physical layer detects the receiving clock from the incoming bit stream through the Clock and Data Recovery (CDR) section and then receives data based on this clock. The transmitter transmits data using a local oscillator with a frequency accuracy of 100 ppm. Synchronization in general Ethernet exists only between two adjacent nodes and is not passed on to other nodes. Data reception in SyncE is same as in conventional Ethernet. Unlike conventional Ethernet, data is transmitted from the sending end on the basis of the clock recovered through the CDR, so that in SyncE, unlike conventional Ethernet, synchronization can be transferred to other nodes. Therefore, in SyncE, before the recovered clock is transferred to the transmitter, a PLL must be used to remove not only the jitter generated in the clock recovery circuit but also the jitter and offset generated in the network. And also to increase the synchronization accuracy, the PLL must have an accuracy of ± 4.6 ppm when operating without a reference clock. Based on this principle, in SyncE, each node recovers clock information from the node located upstream, filters jitter and offset via DPLL, and then distributes it to the nodes located downstream through the physical layer not concerned with the higher layer transport protocols [4]. Fig. 4 shows the timing and data transmitted over a SyncE network through a path.

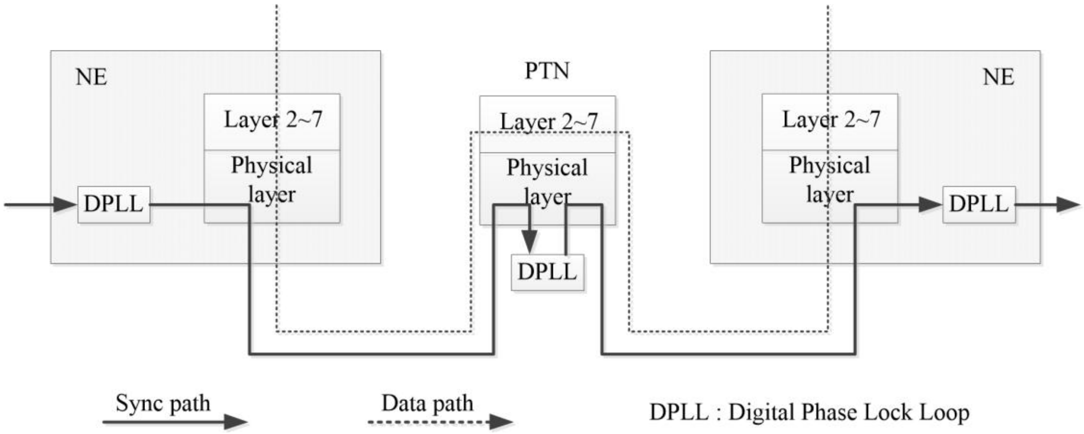


Fig.4. Synchronization and data path in SyncE

In a bit stream being inputted into a network element within the packet transport network, data is processed at all layers corresponding to layer 1 to layer 7 of the open-system interconnection reference model and then transferred back to the next network element, while synchronous information is processed at the physical layer corresponding to layer 1 and is transmitted to the next network element without being transferred to the upper layer. The quality level (QL) is specified to distinguish the clock source according to frequency accuracy and offset [16–19]. The quality level is used to select the highest quality clock source from several clock sources that are fed to the network element. The four classes of quality levels used in SyncE are listed in Table 2.

QL-PRC : The quality level corresponding to the primary reference clock (PRC) defined in ITU-T Recommendation G.811 [19]. QL-SSU-A : is the quality level for the slave clock of type 1 or type 5 defined in Recommendation G.812 [17]. QL-SSU-B : is the quality level corresponding to the slave clock of type 6 defined in Recommendation G.812 [17]. QL-SEC : The quality level corresponding to the synchronous equipment clock (SEC) defined in Recommendations G.813 and G.8262 [12, 16]. QL-DNU : This quality level is not used for synchronization [18]. The network operator must match the quality level for the clock source to one of the four types above. In SyncE, network elements are arranged according to the level of clock resolution, as shown in Fig. 5.

Table 2. Hierarchy of quality level

Quality Level(QL)	Priority
QL-PRC	highest
QL-SSU-A	|
QL-SSU-B	|

QL-SEC	|
QL-DNU	lowest

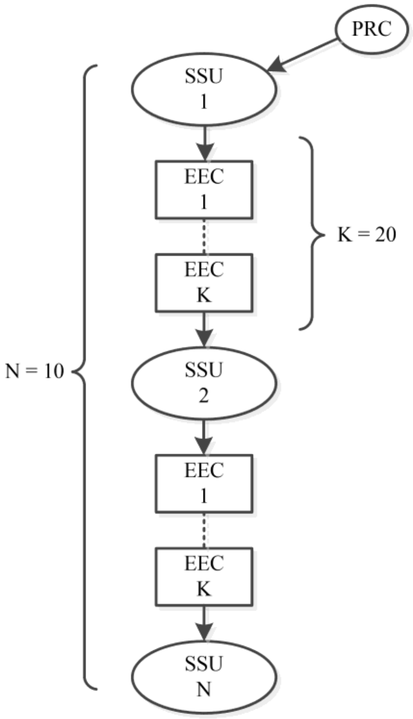


Fig.5. Hierarchical architecture of SyncE

Here, the synchronous Ethernet Equipment Clock (EEC) is a clock quality specification for SyncE and has the same characteristics as SEC for SDH networks. As it can be seen in Fig. 5., the jitter of the clock accumulates because the synchronous network has a master-slave hierarchical structure. Therefore, the EEC connection within one loop should not exceed 20, and then the Synchronous Supply Unit (SSU) should be placed to eliminate the accumulated timing jitter. Within one loop, the SSU should not exceed 10.

3. Design of Synchronous Ethernet-Based Frequency Synchronization System

Synchronous Ethernet technology is a synchronous technology that recovers the clock from the received bit stream at the physical layer level and then passes it to the next stage. To implement SyncE, the functionality shown in Fig. 6 must be implemented at the physical layer level.

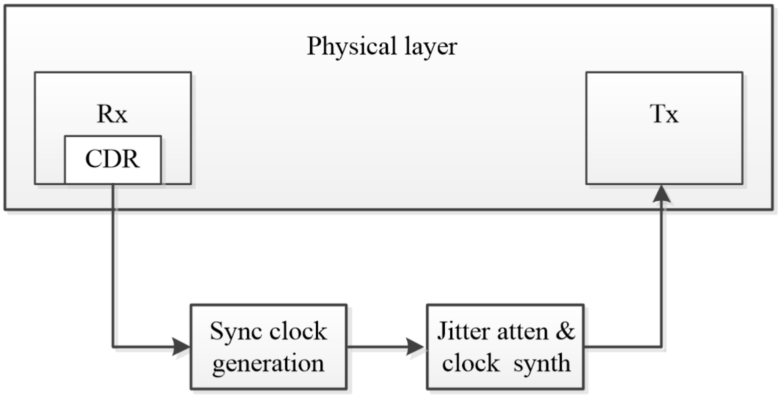


Fig. 6 Implementation principle and hierarchical model of SyncE

First, it is necessary to generate a synchronous clock from the receive recovery clock that is outputted in the clock and data recovery stages. Second, the frequency jitter from the clock and data recovery stage and the synchronous clock generation stage must be eliminated and the clock for transmission must be synthesized. In this paper, a frequency synchronization system is designed to perform the SyncE function for packet transmission equipment with 1G transmission capability used in the access layer of the packet transport network and packet transmission equipment with 10G transmission capability used in the aggregation layer.

3.1. Detection of the Receive Recovery Clock

At the physical layer, the receiving clock is detected at the CDR as shown in Fig. 6. However, in conventional Ethernet, the receive recovery clock is not available externally because it is not outputted from outside the physical layer, and the clock used in the transmit block of the physical layer is input from a fixed local oscillator, thus making it impossible to control the clock externally. For this reason, physical layer devices supporting SyncE are being used [25, 28–30]. The fact that the physical layer chip supports SyncE functionality means that there is a receive recovery clock output and a transmit clock input. Thus, the receive recovery clock can be detected from the physical layer chip supporting the SyncE functionality.

3.1.1. 1G SyncE

The 1G transmission utilizes the physical layer device B50640 to provide SyncE functionality. The B50640 is a 4-channel physical-layer device with a throughput of 1Gbps and has two receive recovery clock outputs [28]. The B50640 can be used to perform pointto-point clock synchronization by outputting the recovered clock from the adversary MDI link (fiber optic or twisted pair). The recovered clock has a frequency of 25 MHz when the device is connected in 1000 BASE-T, 100 BASE-TX, and 100 BASE-FX modes. 10BASE-T is not available for SyncE because it uses Manchester coding and the clock phase information is only available when the packet is transmitted. For the 1000 BASE-T mode, it is necessary to auto-negotiate in the slave mode to output the recovered clock from the adversary link. If the B50640 is self-negotiating in the master mode, it will recover its own clock rather than the clock recovered from the link. The B50640 outputs two recovery clocks (REC CLK[1],REC CLK[0]) and two PLL synchronization signals (LC RECCLK[1],LCRECCLK[0]) for the SyncE application. The recovered clock can be a reference clock of B50640 or a recovered clock from four MDI ports. The port which is the source of the recovery clock can be selected programmable by manipulating the internal register of the chip. The synchronous valid signal LC RECCLK is high when the recovery clock signal REC CLK is synchronized with the incoming signal, and low when it is not synchronized. The reference clock REF CLK input to the physical layer device B50640 can be 125 MHz or 25 MHz depending on the application.

3.1.2. 10G SyncE

10G Equipment consists of a 10G board with two 10G ports, a 1G board with eight 1G ports, and a motherboard, VSC8489 on the 10G board and B50285 on the 1G board, which are used as physical layer devices. The B50285 is an 8-channel 1G physical layer chip with the same SyncE functionality as the 4-channel physical layer chip B50640 [29]. The VSC8489 is a two-channel 10G physical layer chip that supports SyncE functionality, with two receive recovery clock outputs (RXCKOUT), a reference clock input (XREFCLK), and a SyncE reference clock input (SREFCLK) [30]. VSC8489 supports various SyncE configurations in 1G and 10G operating modes. In a SyncE application, at any one time, only one master device must be selected from the internal line side synchronizer or SREFCLK. The supporting SyncE architecture is as follows. Single unit, internal master : In this architecture, the line-side receiver detects the clock signal from the input data and then distributes it to the line-side transmitter of all other ports to achieve synchronous operation. Single-clock, external master. In this architecture, the XREFCLK clock changes slowly with the SyncE clock generated (by) using an external synchronous circuit. The XREFCL source is generated from a recovery clock that is output to the RXCKOUT pin. Dual-clock, external master : In this architecture, the XREFCLK is fed from a fixed 156.25 MHz system clock or crystal. And all the line-side transmitters are synchronized to the SREFCLK clock and output data. In this case, the SREFCLK clock should be 156.25 MHz. We used dual-clock, external master architecture. That is, XREFCLK uses a 156.25MHz crystal oscillator to input a fixed frequency, and SREFCLK inputs a 156.25MHz clock synchronized to the RXCKOUT receive recovery clock output. The frequency of the recovery clock output with the RXCKOUT pin is 161.13MHz.

3.2. Generation of synchronous clocks

Synchronous clock generation is the step of generating a low-frequency clock synchronized to the receive recovery clock output at the physical layer. For the 1G transmission, the receive recovery clock is 25MHz, thus synthesizing an 8kHz synchronous clock. The generation of synchronous clocks in a 10G transmission facility is done in the following manner. The physical layer device (VSC8489 for 10G, B50285 for 1G) generates and outputs a receive clock (161.13MHz for 10G, 25MHz for 1G) from the bit stream that is inputted from the line. On each board, the output to the motherboard produces a synchronous clock of 256KHz synchronized with the receive recovery clock, which generates a synchronous clock of 8KHz on the motherboard again. The details of the clock generation step on the 10G board are as follows. The VSC8489 chip data indicate a recovery clock of 161.13 MHz, but the detailed calculation shows that the line data rate is 10.3125 Gbps and the 64B/66B encoding scheme is used, so the frequency of the recovery clock at the receiver side is 10.3125GHz/64 = 161.1328125 MHz. Therefore, we synthesize a synchronous clock of 25 MHz from this clock. The reason for this is that the clock frequency recovered on the 1G board is 25 MHz, so the same timing block can be used. The frequency dependence is 161.1328125128/825 = 25, which means that the frequency synthesizer synthesizing a 25MHz clock from a 10G physical layer recovery clock must have a multiplication ratio of 128 and a division ratio of 825. In general, the FPGA chip has a built-in PLL, which allows frequency synthesis. However, while the built-in PLL resources are limited by two to four chip codes, the number of PLLs to use is more than this. And also, in the device-supported PLL, it is not possible to synthesize the frequencies corresponding to the multiplication ratio calculated above. Therefore, we perform frequency synthesis using a Numerical Control Oscillator (NCO). The block diagram of the general NCO is shown in Fig. 7.

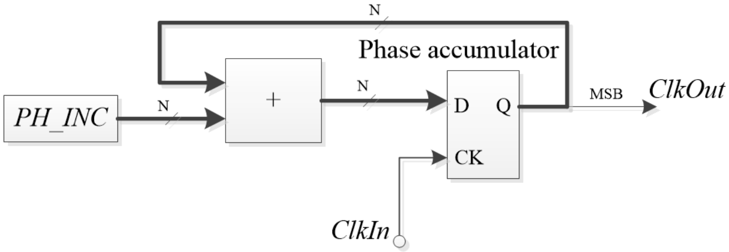


Fig.7. General architecture of clock synthesizer using NC oscillator

In this case, the resultant frequency is

$$ClkOut = \frac{ClkIn}{2^N} \times PH\_INC \qquad (1)$$

where

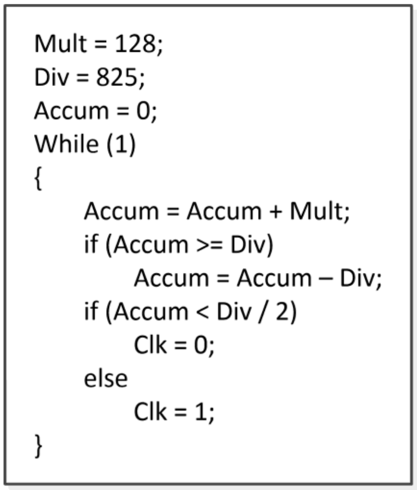
PH_INC - phase increment,
ClkIN - input clock,
ClkOut - output clock.

The amount of phase increment required for the desired frequency synthesis is

$$PH\_INC = \frac{ClkOut}{ClkIn} \times 2^N \qquad (2)$$

Substituting $N = 24$, $ClkIn = 161.1328125$ MHz and $ClkOut = 25$ MHz into Eq. (2), the value of $PH\_INC \approx 2603010$ is obtained. Substituting this value back into Eq. (1), the output frequency is 24.999MHz. However, it cannot be used as a synchronous clock because a frequency of 25 MHz is not obtained exactly. The frequency mixing ratio of 128/825 calculated earlier cannot be obtained by setting N to an arbitrary value. This means that the synthesizer architecture shown in Fig. 7 cannot be used in its entirety.

Recalling Eq. (1), the denominator is an exponent of 2, which means a phase accumulator of N bits. We can now synthesize the clock for any frequency synthesis ratio by making this phase accumulator saturated at 2^N, which is not an exponent of 2, i.e., saturated at the desired division rate. The principle of this frequency synthesis is shown in Fig. 8.



```
Mult = 128;
Div = 825;
Accum = 0;
While (1)
{
    Accum = Accum + Mult;
    if (Accum >= Div)
        Accum = Accum – Div;
    if (Accum < Div / 2)
        Clk = 0;
    else
        Clk = 1;
}
```

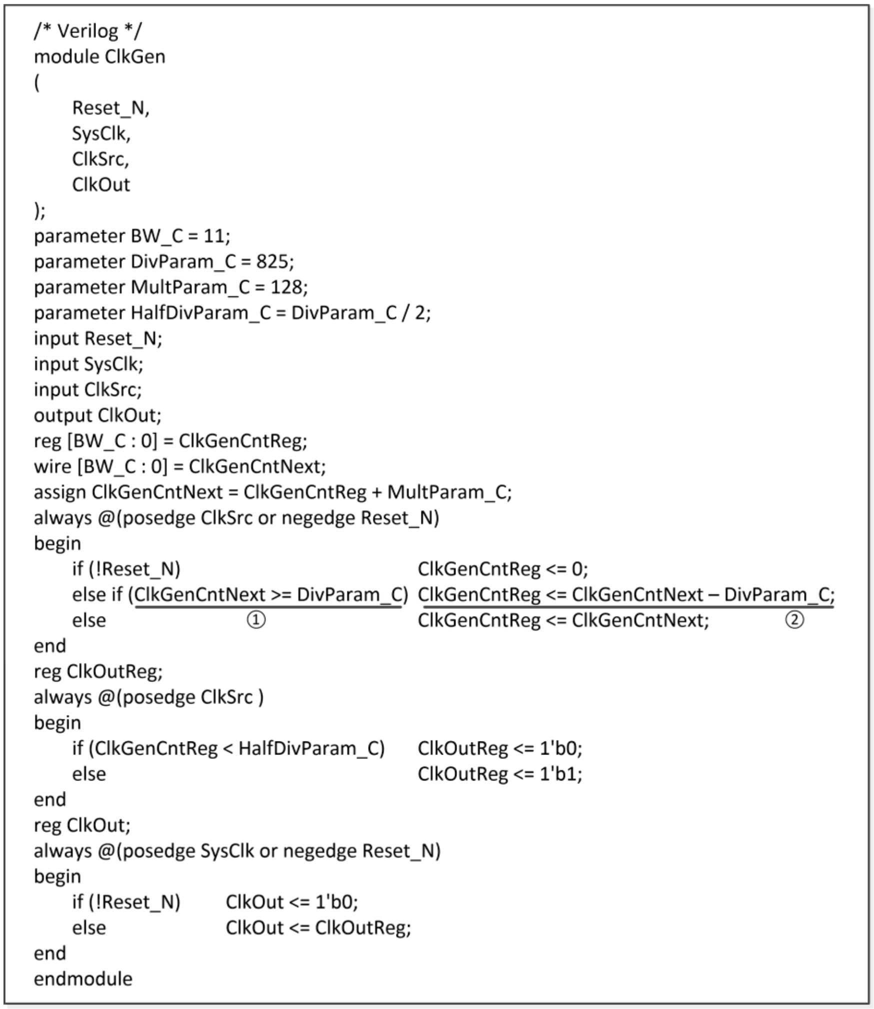
Fig.8. Clock synthesis algorithm with arbitrary frequency synthesis ratio

Here the variables *Mult* and *Div* correspond to *PH_INC* and 2^N in Eq. (1).

Therefore, the resultant frequency is

$$ClkOut = \frac{ClkIn}{Div} \times Mult \tag{3}$$

To implement the frequency synthesis algorithm shown in Fig. 8 on the FPGA chip, a program written in Verilog, a hardware description language, is shown in Fig. 9, and the phase accumulator model among the circuits synthesized using the written program is shown in Fig. 10.



```
/* Verilog */
module ClkGen
(
      Reset_N,
      SysClk,
      ClkSrc,
      ClkOut
);
parameter BW_C = 11;
parameter DivParam_C = 825;
parameter MultParam_C = 128;
parameter HalfDivParam_C = DivParam_C / 2;
input Reset_N;
input SysClk;
input ClkSrc;
output ClkOut;
reg [BW_C : 0] = ClkGenCntReg;
wire [BW_C : 0] = ClkGenCntNext;
assign ClkGenCntNext = ClkGenCntReg + MultParam_C;
always @(posedge ClkSrc or negedge Reset_N)
begin
      if (!Reset_N)                         ClkGenCntReg <= 0;
      else if (ClkGenCntNext >= DivParam_C) ClkGenCntReg <= ClkGenCntNext – DivParam_C;
      else                    ①            ClkGenCntReg <= ClkGenCntNext;          ②
end
reg ClkOutReg;
always @(posedge ClkSrc )
begin
      if (ClkGenCntReg < HalfDivParam_C)    ClkOutReg <= 1'b0;
      else                                  ClkOutReg <= 1'b1;
end
reg ClkOut;
always @(posedge SysClk or negedge Reset_N)
begin
      if (!Reset_N)       ClkOut <= 1'b0;
      else                ClkOut <= ClkOutReg;
end
endmodule
```

Fig. 9. Clock synthesis program written in Verilog language.

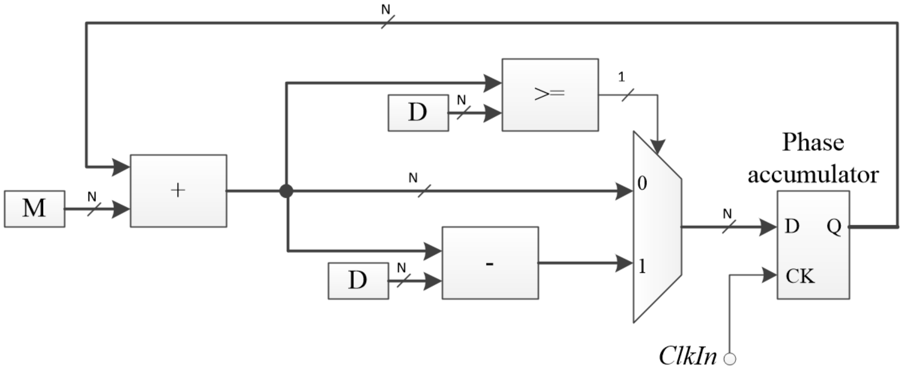


Fig. 10. Synthesized phase accumulator.

In Fig. 10, M stands for body displacement, D stands for division ratio, and M = 128, D = 825, N = 11. The phase accumulator is a sequential circuit consisting of a D flip-flop, the operating clock frequency is 161.1328125 MHz, and the clock period is 6.2 ns. The output of the phase accumulator is fed back through a combinational circuit consisting of an adder, a subtractor, a comparator, and a selector, i.e., an asynchronous circuit, so that the operation result of the combinational circuit must be stable within 6.2 ns of the clock period. For the FPGA chip feasibility, the delay of one adder from the selected chip speed class is 4 ns, so the delay time of two adders is 8 ns, which does not meet the required settling time. Therefore, in the program shown in Fig. 9, the following modifications are made.

3.2.1. Conditional (comparator)

The conditional statement (ClkGenCntNext >= DivParam_C) in Fig. 9 is synthesized as a comparator in the circuit.

Now, changing this conditional simplifies the conditional to

$$(\text{ClkGenCntNext} - \text{DivParam\_C} >= 0) \Longrightarrow$$

$$\Longrightarrow (\text{ClkGenCntReg} + \text{MultParam\_C} - \text{DivParam\_C} >= 0) \Longrightarrow$$

$$\Longrightarrow (\text{ClkGenCntReg} + \text{MultParam\_C}' >= 0) \Longrightarrow$$

$$\Longrightarrow (\text{ClkGenCntNext}' >= 0)$$

where MultParam_C' = MultParam_C - DivParam_C, ClkGenCntNext' = ClkGenCntReg + MultParam_C'.

The conditional statement (ClkGenCntNext' >= 0) is passed to the one-bit comparison of the conditional statement (ClkGenCntNext' [MSB] = 0) since it is only necessary to consider the sign bit of ClkGenCntNext. Therefore, you can remove the conditional and enter the ClkGenCntNext'[MSB] bit directly into the selector.

3.2.2. Subtraction Circuit

Analyzing the synthesized circuit in Fig. 10, an additional subtraction circuit is synthesized since Div must be subtracted from the adder output again when the comparator result is true.

Reviewing this part in Fig. 9 simplifies to

$$(\text{ClkGenCntReg} <= \text{ClkGenCntNext} - \text{DivParam\_C}) \Longrightarrow$$

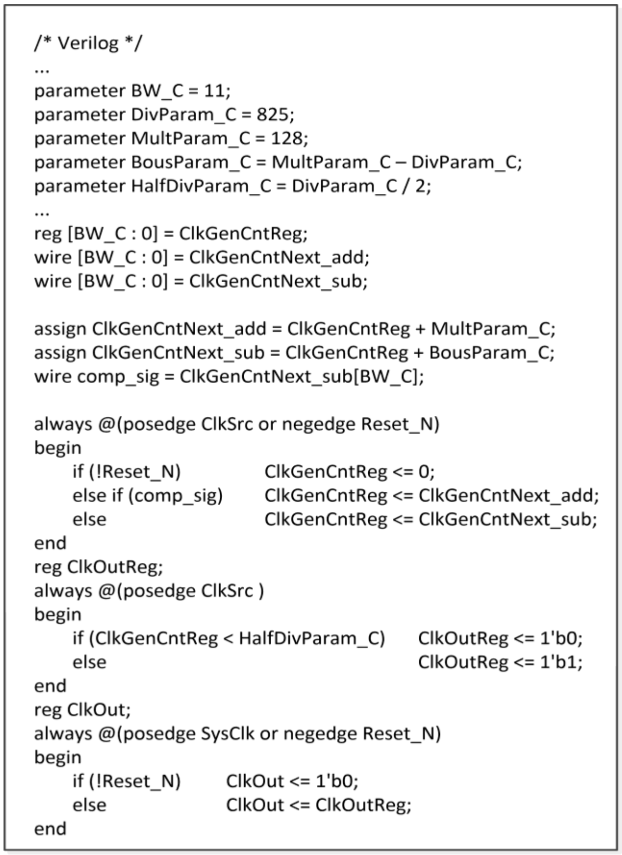
$$\Longrightarrow (\text{ClkGenCntReg} <= \text{ClkGenCntReg} + \text{MultParam\_C} - \text{DivParam\_C}) \Longrightarrow$$

$$\Longrightarrow (\text{ClkGenCntReg} <= \text{ClkGenCntReg} + \text{MultParam\_C}') \Longrightarrow$$

$$\Longrightarrow (\text{ClkGenCntReg} <= \text{ClkGenCntNext}')$$

That is, the additional subtraction circuit is removed and the result of the addition used in the conditional statement can be used in its entirety.

The Verilog program is shown in Fig. 11 and the synthesized circuit in Fig. 12.



```
/* Verilog */
...
parameter BW_C = 11;
parameter DivParam_C = 825;
parameter MultParam_C = 128;
parameter BousParam_C = MultParam_C – DivParam_C;
parameter HalfDivParam_C = DivParam_C / 2;
...
reg [BW_C : 0] = ClkGenCntReg;
wire [BW_C : 0] = ClkGenCntNext_add;
wire [BW_C : 0] = ClkGenCntNext_sub;

assign ClkGenCntNext_add = ClkGenCntReg + MultParam_C;
assign ClkGenCntNext_sub = ClkGenCntReg + BousParam_C;
wire comp_sig = ClkGenCntNext_sub[BW_C];

always @(posedge ClkSrc or negedge Reset_N)
begin
      if (!Reset_N)            ClkGenCntReg <= 0;
      else if (comp_sig)       ClkGenCntReg <= ClkGenCntNext_add;
      else                     ClkGenCntReg <= ClkGenCntNext_sub;
end
reg ClkOutReg;
always @(posedge ClkSrc )
begin
      if (ClkGenCntReg < HalfDivParam_C)      ClkOutReg <= 1'b0;
      else                                    ClkOutReg <= 1'b1;
end
reg ClkOut;
always @(posedge SysClk or negedge Reset_N)
begin
      if (!Reset_N)       ClkOut <= 1'b0;
      else                ClkOut <= ClkOutReg;
end
```

Fig.11. Improved clock synthesis program written in Verilog language.

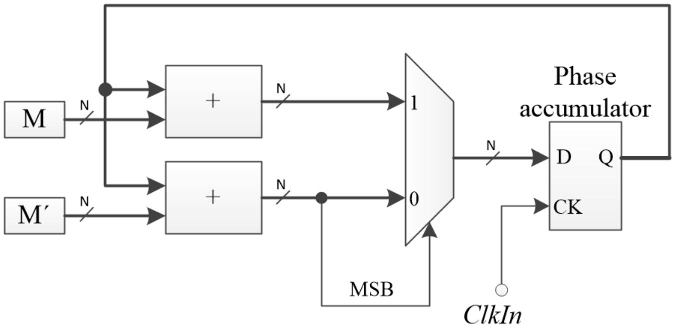


Fig.12. A phase accumulator circuit synthesized using the improved method

In the figure, M = 128, M' = M - D = 128 - 825 = - 697. As can be seen, the input circuit of the phase accumulator is still an asynchronous circuit consisting of a combinational circuit, but the delay time is much shorter because it consists of one adder and a selector, so the circuit can operate correctly.

3.3. Implementation of the PLL

The role of the phase-locked loop (PLL) is to synthesize the synchronous clock for the physical layer device after eliminating the frequency jitter by taking as input the synchronous clock outputted from the synchronous clock generator. For this purpose, an external synchronizer Si5328 was used.

Si5328 is a jitter-attenuating precision clock multiplier for SyncE applications [27]. Si5328 meets all the requirements described in ITU-T Recommendation G.8262 regarding the characteristics of clocks for SyncE. Si5328 generates two output clocks with a frequency range of 8 KHz to 808 MHz from two input signals with a frequency range of 8 KHz to 710 MHz. The internal schematic of Si5328 is shown in Fig. 13.

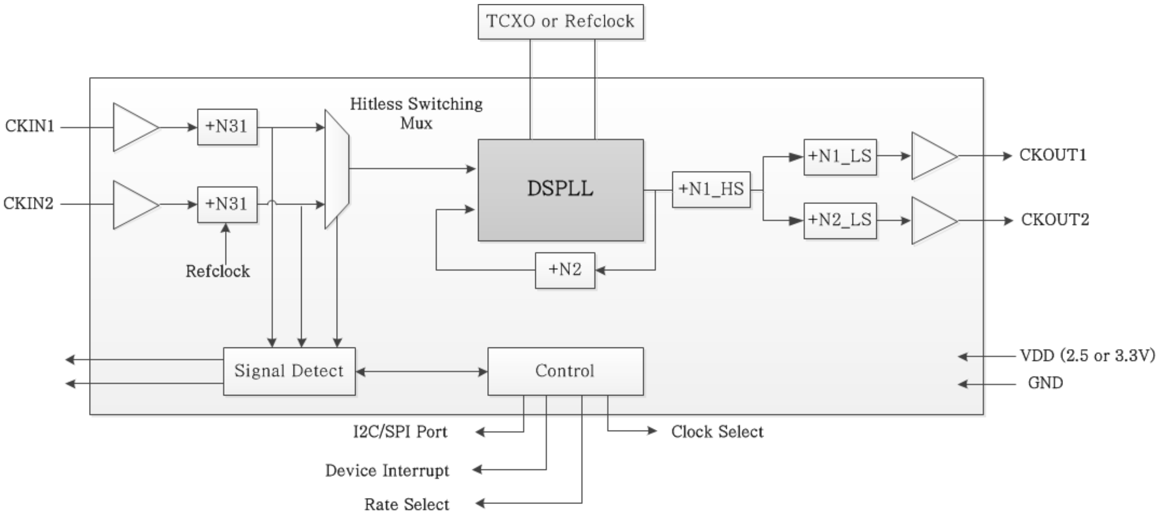


Fig.13. Si5328’s functional block diagram

Two clock outputs are generated from one source. The selection of the clock source used for the synthesis is done with two input clocks, which are automatically checked by checking the validity of the input clock signal and the priority set. Si5328 can also synthesize frequency using TCXO as a clock source. The input clock frequency and the output clock allocations of the Si5328 can be set programmable through I2C or SPI interfaces. In this case, the relationship between the input clock and the output clock is

$$CKOUT1 = \frac{CKIN}{N31} \times \frac{N2}{N1\_HS \cdot N1\_LS} \tag{4}$$

$$CKOUT2 = \frac{CKIN}{N32} \times \frac{N2}{N1\_HS \cdot N2\_LS} \tag{5}$$

where CKIN - input clock frequency,
CKOUT1, CKOUT2 - frequency of output clock 1, 2,
N31, N32 - divide ratio of input divider,
N2 - multiplication ratio of PLL,
N1_HS - divide ratio of frequency synthesizer,

N1_LS, N2_LS - output divide ratio for each output clock.

The loop filter bandwidth of the DPLL is programmable to achieve minimum jitter performance across the range of applications.

3.3.1. 1G Sync Ethernet

The frequency of the input clock is set to 8 kHz, and the frequency of the output clocks are set to 25 MHz and 125 MHz respectively.

CKIN = 8KHz, N31 = N32 = 1, N1_HS = 7, N2 = 93750, N1_LS = 30, N2_LS = 6

$$CKOUT1 = \frac{8}{1} \times \frac{7 \times 93750}{7 \times 30} = 25 \text{ (MHz)}$$

$$CKOUT2 = \frac{8}{1} \times \frac{7 \times 93750}{7 \times 6} = 125 \text{ (MHz)}$$

Here, output clock of 125 MHz is inputted to the physical layer chip and used to synchronize the output bit stream, and output clock of 25 MHz is used to synthesize the external clock.

3.3.2. 10G Sync Ethernet

The frequency of the input clock is set to 8 kHz, and the frequency of the output clocks are set to 156.25 MHz and 25 MHz, respectively.

CKIN = 8KHz, N31 = N32 = 1, N1_HS = 4, N2 = 156250, N1_LS = 8, N2_LS = 8

$$CKOUT1 = \frac{8}{1} \times \frac{4 \times 156250}{4 \times 8} = 156.25 \text{ (MHz)}$$

$$CKOUT2 = \frac{8}{1} \times \frac{4 \times 156250}{4 \times 50} = 25 \text{ (MHz)}$$

Here, output clock of 156.25 MHz is inputted to the physical layer chip and used to synchronize the output bit stream.

4. Performance analysis of frequency synchronization system using SyncE

4.1. Configuration of the test synchronous network

In order to analyze the performance of the frequency synchronization system implemented using SyncE, a test network consisting of six transmissions is constructed as shown in Fig. 14.

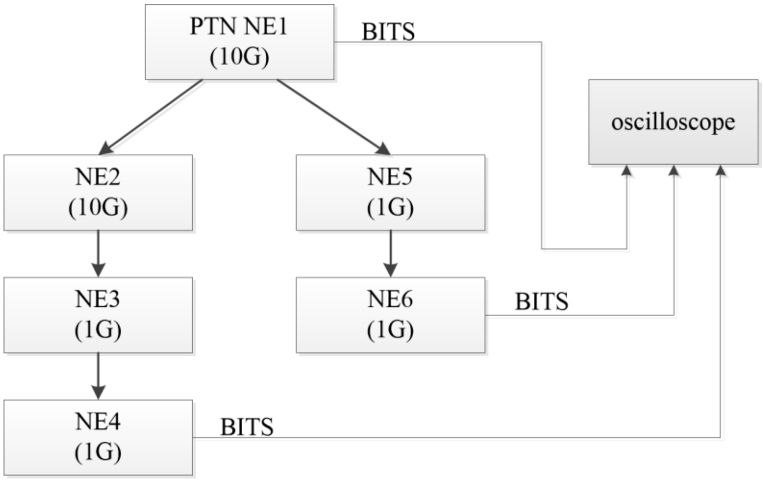


Fig.14. Configuration of the test synchronous network

The detailed connections between the facilities are shown in Fig. 15. Here, HUB, 3110 means the 10G and 1G transmission equipment that implements our synchronization method, and 6110 is the 1G transmission equipment that is already in use. The suffix number represents the equipment number. That is, HUB_1 represents 10G transmission unit 1 and 3110_1 represents 1G transmission unit 1.

The timing accuracy of the 3110_1 and 3110_2 equipment for HUB_1 is measured by setting HUB_1 as the timing source, i.e., the time and frequency sources. As shown in Fig. 15, the connection between HUB_1 and HUB_2 is connected with 10G port, and the connection between the rest of the equipment is connected with 1G port.

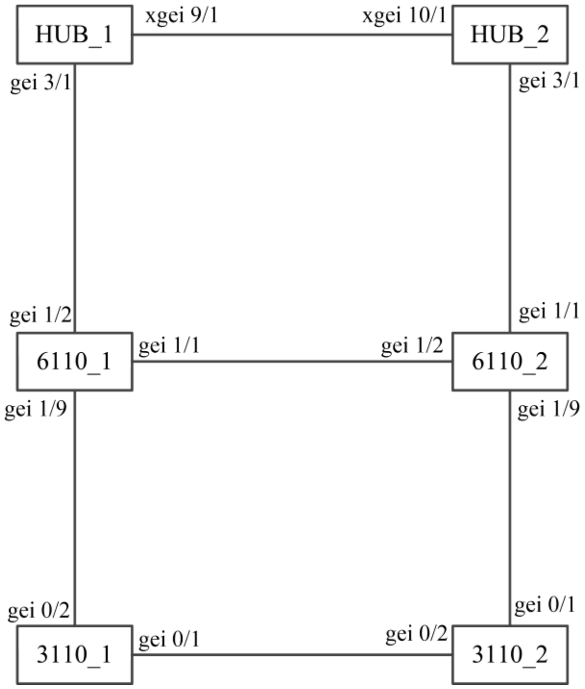


Fig.15. Connection status of equipment in the test network.

4.2. Synchronization Measurements

The timing measurement is performed by measuring the clock synchronized with the reference clock and the reference clock on an oscilloscope to evaluate the timing status.

The oscilloscope with two measuring channels inputs a reference clock to channel 1 and a clock synchronized to the reference clock to channel 2.

The oscilloscope is operated in trigger mode and the trigger source is set to measurement channel 1. Since channel 1 is set as the trigger source, the waveform displayed in channel 1 is always stopped. If the clock input to channel 2 is not synchronized with the clock input to channel 1, the waveform displayed in channel 2 will flow.

When the clock input to channel 2 is correctly synchronized with the clock input to channel 1, all waveforms displayed in channel 1 and channel 2 are stopped. In this state, the oscilloscope waveform recording mode can be set to the infinite mode to measure the frequency accuracy in the synchronized state.

The timing measurement in the synchronous test network (Fig. 14) is performed as described above. Here, the synchronous source is set to the internal clock of transmission unit 1 at the highest level. The reason for this is to configure the test network as transmission facilities simply.

Transmission facilities at the lower level are supplied with synchronization from transmission facilities at the upper level. Transmission Equipment 1 and Transmission Equipment 2 are configured to ensure that the 10G SyncE function is functioning properly. The ultimate goal of the test synchronization network is to ensure that the transmission equipment 4 and 6, which are at the lowest level in transmission equipment 1, are correctly synchronized with the synchronous source. The clock used for the measurement is the BITS output clock on each unit. The BITS clock is set in 2 MHz mode.

4.3. Measurement results and performance analysis

The measurement results of the transmission equipment 1 BITS output of the test synchronous network (Fig. 14) connected to the measurement channel 1 of the oscilloscope and the BITS output of the transmission equipment 4 to the measurement channel 2, respectively, are shown in Figs. 16, 17 and 18, respectively, with the measurement method described above.

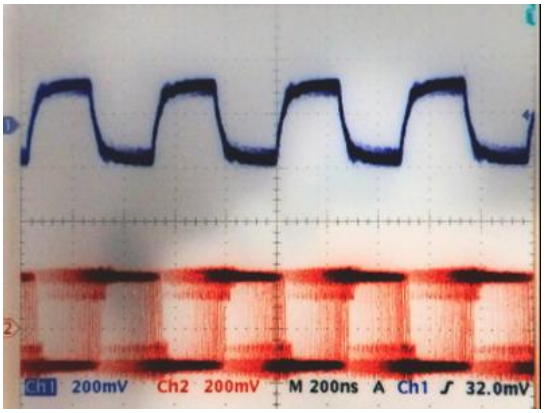


Fig.16. BITS output waveform in the unsynchronized state

As shown in Figs.16 and 17, it can be seen that the waveform displayed on channel 2 flows if the clock input to measurement channel 2 is not synchronized with the reference clock input to measurement channel 1, and stops when it is synchronized with the reference clock.

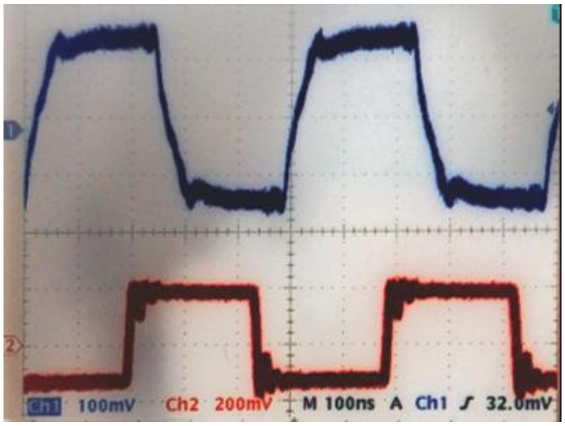


Fig.17. Synchronous BITS output waveform

Fig.18 shows the waveform recorded by the oscilloscope in the infinite mode and measured over a long period of time (3 days) while the clocks entering the measurement channels 1 and 2 are synchronized. As can be seen from the waveform, the phase fluctuation does not exceed 23 ns. This fully meets the requirements of SyncE[20].

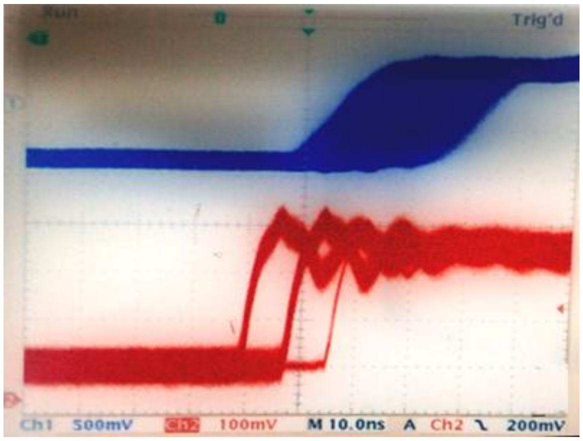


Fig.18. clock jitter in synchronized state

Fig.19 shows the synchronization state of network elements in the U31 network management system, where the arrows show the synchronization direction.

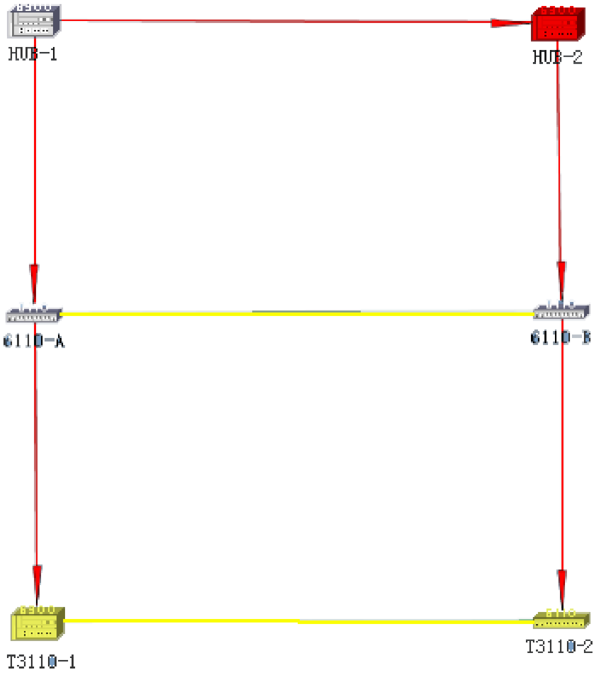


Fig.19. Synchronous direction of the test network

Hence, 6110_1 and HUB_2 receive synchronization from HUB_1, 3110_1 from 6110_1, 6110_2 from HUB_2, and 3110_2 from 6110_2. Finally, it can be seen that the synchronization network in Fig. 14 is completely consistent.

5. Conclusion

In this paper, we describe the design and implementation of a frequency synchronization system, which is one of the important functions in packet transport networks. Synchronous Ethernet-based frequency locking system is composed of a synchronous clock generation unit, a jitter cancellation and a frequency doubling unit.

The synchronous clock generation unit detected the receive recovery clock from the receive bit stream that was fed to the physical layer, and then produced a synchronous clock synchronized to this clock. In 1G SyncE, a receiver recovery clock rising edge of 25 MHz is detected to synthesize an 8 KHz sync clock synchronized to this edge. In 10G SyncE, instead of using the PLL from the receive recovery clock 161.1328125 MHz, a 25 MHz SyncE clock is generated using the NCO, and then the 8 KHz SyncE clock is synthesized in the same way as for 1G SyncE.

In the jitter cancellation and frequency doubling stages, an external synchronization loop is used to remove the phase jitter generated by the receive recovery clock detection stage and the synchronous clock generation stage from the synchronous clock, and to synthesize the required frequency for SyncE. The parameters of the external synchronization loop were set to synthesize a 125 MHz clock from an 8 KHz clock on 1G SyncE and a 156.25 MHz clock from an 8 KHz clock on 10G SyncE. A synchronous test network consisting of two 10G transmission units and four 1G transmission units was constructed, and the BITS output clock at 2 MHz was measured and analyzed on an oscilloscope. The phase jitter of our synchronous system is 23 ns, which meets the requirements specified in SyncE.

Acknowledgements. The authors would like to thank the editor and anonymous reviewers for valuable comment and advice.

Declarations

- Funding

The authors received no specific funding for this study.

- Conflict of interest/Competing interests

The authors declare that they have no conflicts of interest to report regarding the present study.

- Ethics approval and consent to participate

This article is original and contains unpublished material. The authors testify that this paper submitted to the Circuit, systems and signal processing has not been published elsewhere and that has no ethical issues.

- Consent for publication
- Data availability

All data generated or analyzed during this study are included in this published article and its supplementary information files.

• Materials availability • Code availability

- Author contribution

Un Chol Song : Participated in all experiments, coordinated the data collection and analysis and contributed to the writing of the manuscript.

Jong Sam Ri : Concept development, data validation and contributed to review the article critically and gave the final approval of the version to be submitted.

REFERENCES

[1] T.R.Hsing and K.N.Lau, “Networks and Services: Carrier Ethernet, PBT, MPLS-TP, and VPLS,” John Wiley & Sons, pp.303-332, 2012.

[2] ETSI-TS 125 104, “Universal Mobile Telecommunications System (UMTS); Base Station (BS) Radio Transmission and Reception (FDD) (3GPP TS 25.104 version 8.6.0 Release 8),” Mar. 2009.

[3] ETSI-TS 125 402, “Universal Mobile Telecommunications System (UMTS); Synchronization in UTRAN Stage 2(3GPP TS 25.402 version 8.0.0 Release 8),” Jan. 2009.

[4] M. Gilson and S. Taylor, “SyncE & PTP — Deployment Perspective,” NIST WSTS, Mar. 2010.

[5] R. Megino, “T-Mobile view on Next Generation Mobile Network Synchronization,” ITSF, 2009.

[6] J. L.Ferrant, “Evolution of the Standards for Packet Network Synchronization,” IEEE Commun.Mag., pp.132-138, Feb.2011.

[7] J. L.Ferrant et al., “Synchronous Ethernet: A Method to Transport Synchronization,” IEEE Commun. Mag., Sept.2008.

[8] L.Jeremy, “Hybrid Mode Synchtronous Ethernet and IEEE 1588 in Wireless TDD Applications,” NIST WSTS, Mar. 2010.

[9] ITU-T G.8260, “Definitions and terminology for synchronization in packet networks,” 2010.

[10] ITU-T G.8261, “Timing and Synchronization Aspects in Packet Networks,” Aug. 2013.

[11] ITU-T G.8261.1, “Network requirements for frequency synchronization,” 2011.

[12] ITU-T G.8262, “Timing Characteristics of Synchronous Ethernet Equipment Slave Clock (EEC),” July 2010.

[13] ITU-T G.8265, “Architecture and requirements for packet based frequency delivery,” 2010.

[14] IEEE Std 1588™-2008 “IEEE Standard for A Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” July 2008.

[15] A. Magee, “Synchronization in Next-Generation Mobile Backhaul Networks,” IEEE Commun. Mag., pp.110-116, Oct. 2010.

[16] ITU-T Rec. G.813, “Timing Requirements of SDH Equipment Slave Clocks (SEC).”

[17] ITU-T G.812, “Timing Requirements of Slave Clocks Suitable for Use as Node Clocks in Synchronization Networks.”

[18] ITU-T G.810, “Definitions and Terminology for Synchronization Networks.”, Aug.1996

[19] ITU-T G.811, “Timing characteristics of primary reference clocks”, Sept.1997

[20] ITU-T G.823, The control of jitter and wander within digital networks which are based on the 2048 kbit/s hierarchy, 2000.

[21] RFC 5905, Network Time Protocol Version 4: Protocol and Algorithms Specification, June 2010.

[22] RFC 1305, Network Time Protocol (Version 3) Specification, Implementation and Analysis, March 1992.

[23] RFC 2030, Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI, Oct 1996.

[24] RFC 4330, Network Time Protocol - NTPv4 -(Draft 8) - SNTPv4, 2006.

[25] Reinhard Exel, Menber, IEEE, Thomas Brigler, and Thilo Sauter, “Asymmetry Mitigation in 100BASE-TX Ethernet for High-Accuracy Clock Synchronization” IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 63, NO. 3, MARCH 2014.

[26] F. Lamonaca and D. Grimaldi, “Trigger realignment by networking synchronized embedded hardware,” IEEE Trans. Instrum. Meas., vol. 62, no. 2, pp. 38–49, Jan. 2013.

[27] “Si5328: ITU-T G.8262 Synchronous Ethernet Jitter-Attenuating Clock Multiplier,” Silicon Laboratories, 2013

[28] “B50640E-DS03-R:Quad Gigabit Ethernet Transceiver, ” BroadCom, Oct. 2012

[29] “B50285-AN101-R : Hardware Design Guide,” April 2014.

[30] “VMDS-10416 : VSC8489-10 Dual Channel WAN/LAN/Backplane RXAUI/XAUI to SFP+/KR 10GbE SerDes PHY with VerTime," Vitesse, May 2014.

[31] “CYIV-5V1-2.2 : Cyclone IV Device Handbook,” Altera Corporation, March 2016