100Mbps Ethernet Data Transmission over SDH Networks using Cross Virtual Concatenation

Sunanda Manke, EL & Comm. Engg. Deptt., BUIT, Barkatullah University, Bhopal.
Kavita Khare, EL & Comm. Engg. Deptt., Maulana Azad National Institute of Technology, Bhopal.
S.D. Sapre, EL & Comm. Engg. Deptt., Maulana Azad National Institute of Technology, Bhopal

***Abstract*- Cross Virtual Concatenation is the new technique proposed for bandwidth efficient transmission of data over SDH networks. SDH networks came into existence for reliable voice transmission. As the demand of data traffic grew in wide area networks, new technologies were developed and standardized by ITU for data transmission over SDH networks. The technologies used namely GFP (generic framing procedure), VCAT (virtual concatenation) and LCAS (link capacity adjustment scheme) enable network operator to provide integrated voice and data services over their legacy SDH infrastructure. Data packets are encapsulated using framing protocols GFP. VCAT is a process of distributing the GFP framed data payload in number of virtual channels of same capacity forming a Virtually Concatenated Group (VCG). LCAS is used for dynamic bandwidth allocation. LCAS enhances the VCAT scheme with hitless in service addition and removal of VC's to/from the VCG.**

VCAT combines homogeneous virtual containers(VC's) together which in some cases limits the performance of VCAT. This paper describes the implementation of new concatenation technology named cross virtual concatenation (CVC), which combines heterogeneous VC's together to utilize the SDH bandwidth more efficiently. CVC implementation requires only end node equipments to be upgraded as VCG members travel through the link similar to the conventional VCAT. In this paper transmitter and receiver circuits are designed to transmit and receive 100Mbps Ethernet data using CVC, where two types of VC's namely VC-3 and VC-12 are used for data transmission. The functionality of the circuit is tested using Modelsim Simulator using VHDL. Total Transmission delay is calculated as 125us and proved that, there is no complexity added at the receiver side due to this delay. The receiver is designed for 32ms differential delay compensation which can be increased up to maximum 256ms by increasing the buffer size at the receiver.

***Key Words* :CVC-Cross Virtual Concatenation, GFP-Generic Framing Procedure, ITU – International Telecom Union, LCAS-Link Capacity Adjustment, Scheme, SDH-Synchronous Digital Hierarchy, STM-Synchronous Transport Module, VC-Virtual Channels, VCAT-virtual Concatenation, VCG-Virtually Concatenated Group, VHDL-VHSIC Hardware Description Language.**

I. INTRODUCTION

SDH is the dominant optical transport technology, initially designed and optimized for voice transmission over optical networks. It is a TDM based technology uses bandwidth hierarchy, indicated by STM-n where n=1, 4, 16, 64. The basic unit is STM-1 (155.54Mbps) which supports different types of smaller payloads namely VC-11,VC-12,VC-2 ,VC-3and VC-4 [1].As the traffic demand grew for data transmission in wide area networks, most feasible solution was transmitting data using SDH backbone network infrastructures which were already laid. Technologies were developed and standards were formulated for the same. Fig 1 shows transmission of data using SDH links [5], [9], [10], and [11].

Data packets are encapsulated using framing protocols like GFP and LAPS.GFP is most preferred and commonly used protocol [2], [8]. Second step is mapping the framed data into SDH frame. The technique used for mapping is concatenation. Concatenation is the process of integrating several containers to a unified one to provide a larger tunnel for data transportation. There are two kinds of concatenation, contiguous concatenation (CCAT) and virtual Concatenation (VCAT).

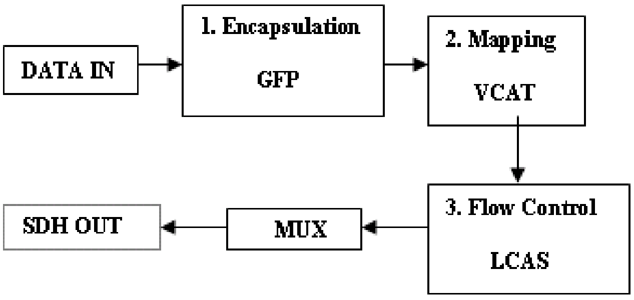


Figure1. Data transmission over SDH

From the perspective of technological and economical feasibility VCAT is preferred [5]. VCAT is a process of distributing the GFP framed data payload in number of virtual channels of same capacity forming a virtually concatenated group(VCG) [1].These VC's may or may not be a part of same STM stream i.e. each channel can independently travel through different links and receiver combines all members of VCG. Receiver compensates for the differential delay introduced between different VC's as they reach destination at different point of times. It stores all the incoming streams and checks for Multi Frame Indicator (MFI) and Sequence number to combine the VC's to extract the data. Flow control is required for dynamic bandwidth allocation as the data rate can vary. The protocol used for flow control is LCAS (link capacity adjustment scheme) [3], [7]. LCAS enhances the VCAT scheme with hitless in service addition and removal of VC's to/from the VCG. Additionally LCAS provides load sharing protection by dynamically removing failed members from the VCG when

978-1-4244-3805-1/08/$25.00 ©2008 IEEE ICON 2008

they experience fault and as the fault is repaired, the member can again be added to achieve the normal flow [3]. After LCAS, signal goes to MUX block where it is multiplexed with other containers/STM's for transportation through SDH link.

VCAT utilizes SDH bandwidth efficiently, but using same type (capacity) of virtual containers limits the performance of VCAT. Cross virtual concatenation (CVC) is combining VC's of different capacity in one VCG to harvest SDH bandwidth more efficiently [4]. This paper describes concept and advantages of CVC in section II. Section III describes the design aspects and hardware implementation of the Transmitter and Receiver circuits for 100Mbps Ethernet data transmission and Reception over SDH using CVC. It also discusses the delay problem and calculation of the minimum delay required in the transmission. Section IV covers the simulation results of the transmitter and receiver circuit. The coding for the hardware is done in VHDL and the functionality of the hardware is verified using Modelsim.

II. Cross Virtual Concatenation (CVC)

Standard VCAT divides the data payload into VC's of same type and capacity. For example transmission of 100Mbps can be achieved either through 46 VC-12 streams or using 3 VC-3 streams. If VC-12 containers are used then maintaining so many circuits is cumbersome process or if we go for 3 VC-3 channels then bandwidth utilization (3*48.38Mbps= 145.15Mbps) is poor. The possible solution could be using 2VC-3 Channels and 2 VC-12 channels (2*48.38+2*2.176=101.12Mbps) in one VCG to transmit 100 Mbps Ethernet data. Such heterogeneous combination of VC's in one VCG is termed as CVC, which is done to achieve greater bandwidth utilization and reduce the circuit complexity.

Table 1 shows the improvement in the bandwidth efficiency obtained for different types of data rates using CVC compared to conventional VCAT.

TABLE 1 BANDWIDTH EFFICIENCY USING VCAT AND CVC

services	Efficiency with Standard VCAT(VC)[6]	Efficiency with Cross VCAT(CVC)
Ethernet (10 Mbit)	VC-12-5v --> 92%	VC-12-5v --> 92%
Fast Ethernet (100 Mbit)	VC-12-46V-> 98% VC-3-3V->69%	VC-3-2V & VC12-2V-> 100%
ESCON (200 MByte)	VC-3-4v --> 98%	VC-3-4V & VC12 -2V-> 100%
Fibre Channel (1 Gbit)	VC-4-6v --> 89%	VC-4-6V & VC-3-2V-> 98%
Gigabit Ethernet (1Gbit)	VC-4-7v --> 85%	VC-4-6V &VC-3-2V -> 98%

CVC needs hardware up gradation at the end nodes only, as VCG members travel through the link similar to the conventional VCAT. It has also been proved that CVC provides greater flexibility in terms of path selection and reduces miss probability in connection establishment in comparison to standard VCAT [4].

III. Implementation

CVC based 100Mbps data transmission over SDH can be achieved using combination of two types of virtual channels namely VC-3 2V and VC-12 2V.The transmitter circuit inputs GFP framed 100Mbps Ethernet data and puts this data into VC's frames one by one, overhead bytes are then calculated and four streams(2 VC3 and 2 VC12) are transmitted finally. These streams can either travel independently through the network or can be put into STM's for transmission.

Receiver circuit receives these four streams and stores them into buffers for differential delay compensation since all the streams follow different paths and reach the destination at different times. The circuit checks for the frame count and sequence number for synchronization and finally outputs 100Mbps Ethernet data by selecting one of the four streams at a time using multiplexer.

A. *Transmitter*

The block diagram of the transmitter is shown in fig 2.Din is the input 100 Mbps stream. Reset signal is used to start the process.clk1 is used to input the data into buffers ,so its data rate should be equal to the input data rate.VC-3 buffers output the data at clk2 rate and VC-12 buffers output the data at clk3 rate.

1. Controller - This block is used to split the load between different buffers. It generates write and reads signals for all the buffers. It consists of three counters. Counter1 counts types of VC's used (here it is two-VC-3 and VC-12). Counter2 counts number of bytes to be put in VC-3 containers [here 756*2=1512, as there are 2 VC-3 containers and frame size for VC-3 is 765 bytes (756 payload +9 overhead bytes)]. Counter3 counts number of bytes for VC-12 containers[34*2=68,as there are two VC-12 containers used and frame size is 35 bytes (34 payload +1 overhead byte)].Initially VC-3 container are enabled one by one using wr1 and wr2 signals and then VC-12 containers are enabled using wr3 and wr4 signals. This process repeats enabling any one buffer at a time to input the data DIN which is applied to all the buffers. Only one buffer is enabled at a time using WR signal. RD signal goes to all the buffers, when it is enabled, all buffers start giving output.

2. VC-3 Buffers – There are two VC-3 buffers used in this circuit as VC3 -2V is required for transmission. Each buffer inputs the data when respective WR signal is enabled. RD signal enables outputting the data. Buffer size taken is twice that of the frame size (756) & calculated as 1512 for each buffer. This is to achieve perfect synchronization for input and output side and avoid any data loss. The frame consists of 9 overhead bytes, which are also calculated before outputting the data and total frame of size 765 bytes is outputted. Sequence number and MFI are also calculated as per the standard and inserted in the output stream at specified places.

3. VC-12 buffers -Two VC-12 buffers are used as per the bandwidth requirement of the input stream. The buffer size taken is 34*4=136 bytes, which is the payload size of one multiframe consisting four frames. As overhead bytes are calculated for the whole multiframe, this buffer size enables calculation of overhead bytes easier. Data is written in to the buffers if corresponding WR signal is enabled and as before data is read out when RD signal is activated. Overhead bytes (including MFI and sequence numbers) are calculated and read out for specific timeslots.

Thus this circuit maps 100Mbps Ethernet data into SDH frame using CVC.

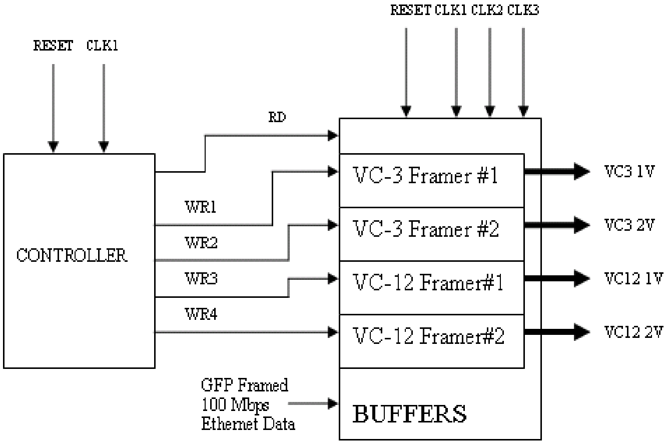


Figure 2 Transmitter block Diagram

B. *Transmission Delay*

Each buffer is a FIFO queue, where data is written at the end of the queue and read from the beginning of the queue, buffer size taken here is double the frame size of each VC to ensure exclusion of read and write memory operation. All buffers are filled sequentially, starting with VC-3 buffer#1, VC-3 buffer2#, VC-12 buffer#1 and VC-12 buffer#2.The number of bytes filled in each buffer is equal to its frame size.756 bytes are filled in VC-3 buffers and 34 Bytes are written in VC-12 buffer. Some amount of delay is required in outputting the data to ensure perfect synchronization and no data loss. This delay is achieved by programming the controller to activate the RD signal only after all the buffers are written once. As buffer size is double to that of the frame size, when data is read from the first half of the buffer it is written in the second half and vice versa. Thus perfect synchronization is achieved and there is no overlapping in reading and writing operation of the memory. The minimum transmission delay can be calculated by adding the time required to fill all the buffers once.

Writing speed = 101.12 Mbps (GFP framed Ethernet data speed)

Total buffer size = 756*2 + 34* 2=1580bytes.

Time required to fill all the buffers once = 1580 * 8 *1/101.12 Mbps= 125us

Thus the transmission delay is equal to one frame size i.e. 125 us. The controller is programmed to generate 125 us delay in RD signal after the WR signal is enabled. As the total stream is delayed by the same amount, there is no complexity added at the receiver side also there is no data loss.

C. *Receiver*

Fig 3 shows the block diagram of the receiver circuit. Two VC-12 streams and two VC-3 streams are input to the circuit and DOUT is the 100Mbps Ethernet output data.

1. Buffer block-There are four buffers used to store the incoming streams. The buffer size depends on the differential delay to be compensated

As per the standard defined, for 256 ms differential delay compensation [1], the buffer size required for individual buffer is calculated as follows.

For VC-3 stream frame count is transmitted in H4 overhead byte of the VC-3 frame[1].There are two multiframe indicators defined(MFI-1 and MFI-2).MFI-1 varies from 0 to 15 and MFI-2 varies from 0 to 255 thus total number of frames are 255 x 16 = 4096 frames[1]. Each frame contains 765 bytes so the total buffer size required is 4096 x 765 = 3133440 bytes.

For VC-12 stream the frame count is transmitted in the 2nd bit of K4 byte of the VC-12 multiframe (which consists of four frames each of size 35 bytes). frame count varies from 0 to 32 and total 32 K4 bits are transmitted before the next frame count. Thus the total buffer size required is 32x32=1024 multiframes i.e. 1024x4x35=143360 bytes.

In this design, we are considering 32ms differential delay, the buffer size for it is calculated as follows.

Each frame is transmitted in 125us ,which means for 32ms differential delay compensation, number of frames will be 32ms/125us= 256 frames, so the buffer size for VC-3 frame will be 256x756 =19584 bytes.256 frames will result in the VC-12 buffer size of 256x35 =8960bytes.

Buffers input the data, store it and extract the frame count and sequence number with the help of WR signal which indicates the starting of the frame. The frame count and the sequence number are then given to the controller block. When controller block gives EN signal, the buffer block starts outputting the data from the location specified by the controller block at ADD signal. If controller gives RST signal (sync failure), the buffer starts from the initial location. Power on RESET is also input to the buffer blocks to initialize different signals used by the buffer block.

2. Controller-controller block is used for synchronization. It takes frame count and sequence number for all the streams from the buffer block and checks for the first frame count of each buffer. When it gets all the counts correctly, it gives EN and ADD signal to buffers to indicate the starting of the frame. All the buffers output the data after receiving EN signal. Controller checks and compares other frame counts and disables EN signal whenever there is a mismatch and gives RST to indicate synchronization failure.

When synchronized, this block also generates RD and ENA signal for data generator, which outputs one of the four inputs depending on the value of RD signal, which varies according to the sequence number.

*3. Data Generato*r -This block consists of buffers and a multiplexer. Four buffers are used to store the four synchronous incoming streams and output them according to the RD signal. RD signal has two bits, so at any instant one of the inputs goes to the output side. To avoid data loss, buffer size is taken as the double of the frame size. For the VC-3 streams, buffer size is 765x2=1530 bytes and for VC-12 streams buffer size is 35x2=70bytes.RD signal goes to a multiplexer inside the generator block which selects one of the buffer and outputs the data from that buffer. For perfect synchronization and to avoid data loss, outputting the data is started only after all the buffers are half filled (one frame size); controlled by the ENA signal generated by the controller block.

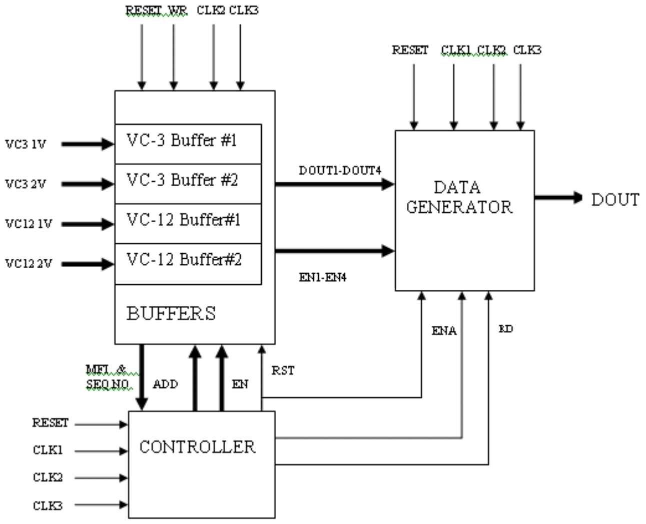


Figure 3 Receiver block Diagram

IV. Simulation Results

A. Transmitter

For simulation purpose, clk1, clk2 and clk3 clock periods are defined as follows

Clk1 = 79ns (101Mbps) Input Data Rate

Clk2 = 0.16us (48.960Mbps) VC-3 Output data rate

Clk3 = 3.57us (2.240Mbps) VC-12 Output data rate

Ethernet data (Din) is simulated using a PRBS generator, which takes CLK input and generates pseudo random signal at the same CLK rate Here CLK signal rate is taken as CLK1, so PRBS block generates random data byte by byte at the Ethernet data rate. PRBS generated Din is applied to the transmitter block and the following results are observed which proves the functionality of the circuit designed.

Fig.4 shows the block diagram of PRBS generator, where CLK and Reset are input signals and Q (parallel 8 bit data) and P1-P8 (Serial data) are output signals. Fig.5 gives the timing waveform obtained from this generator, which then is applied to the transmitter circuit.

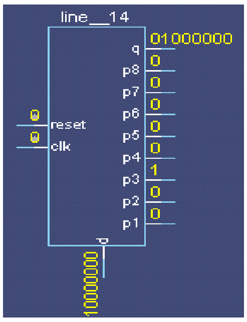


Figure 4 PRBS Generator

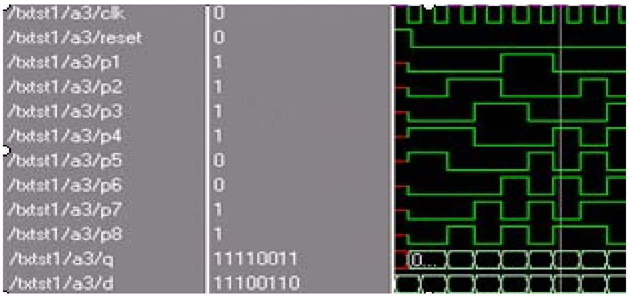


Figure 5 PRBS Generator Timing Diagram

Fig 6 shows the block diagram of the controller block, which takes reset and CLK signal as input and generates WR and RD signal based on the counter values of the three counters. The waveforms of the block in figure 7 shows RD and WR signal waveforms, at any instant only one of the WR signal is enabled so that only one buffer is enabled .RD signal is activated after one complete write cycle which is indicated by marker.

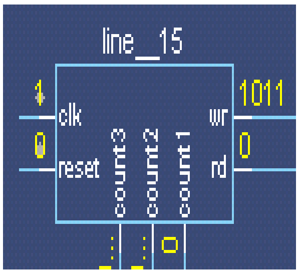


Figure 6 Controller Block

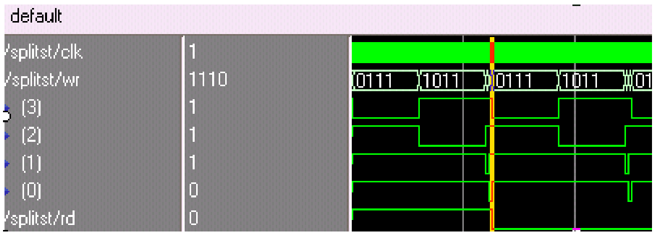


Figure 7 Controller Waveform

Fig 8 shows the block diagram of VC-3 buffer block; dout takes one of the overhead bytes data or data from the memory depending on the position of the byte in the frame. Fig 9 shows that when RD signal goes low, data is read from the buffer i.e. we get dout when RD is enabled.

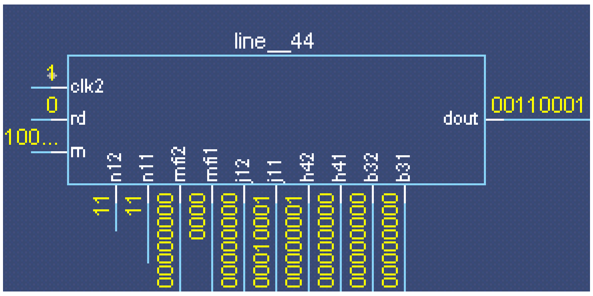


Figure 8 VC-3 Buffer Block

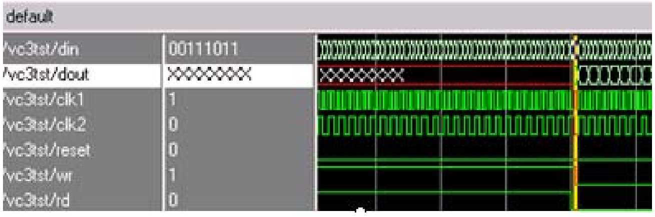


Figure 9 VC-3 Block Waveforms

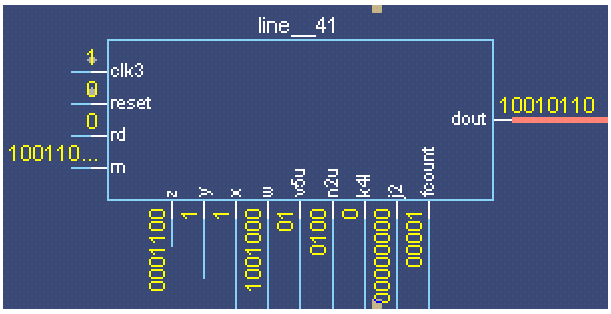


Figure 10 VC-12 Block

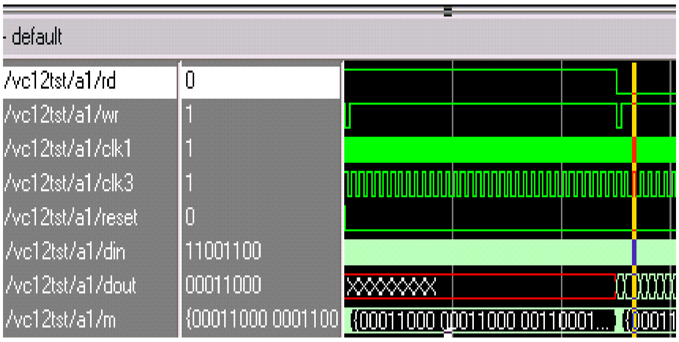


Figure 11 VC-12 Input Output Waveform

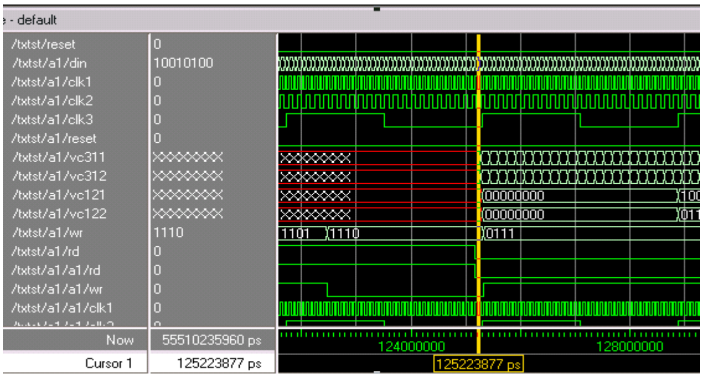


Figure 12 Transmitter Waveform

Fig 10 shows the block diagram of VC-12 block. Fig 11 shows the input output waveform of VC-12 .Fig 12 gives the input and output waveforms of the whole transmitter block, where din is the Ethernet data and four output streams are two VC-3 and two VC-12.data outputs starts after a delay of 125 us when the RD signal is activated. Waveforms also show the different clock signals and other control signals like reset and WR.

B. Receiver

For simulation of the receiver circuit, as before four input streams are simulated using PRBS generator.2 VC-3 streams at CLK2 rate and 2 VC-12 streams at clk3 rate. All signals are then observed based on these input signals using modelsim simulator.

Fig 13 and 14 shows the block diagrams of the controller. Controller takes input signal on d1, d2, d3 and d4 lines (frame count, Sequence number), stores them in m1, m2, m3, m4. It compares the frame count of all the incoming streams and generates address for buffers on add lines. These addresses indicate the starting of the data. Once synchronized, this block gives RD and ENA signal to data generator block, which finally outputs the data. Fig 15 shows the waveform generating the ADD signal based on the contents of memory indicating frame count. Fig 16 shows the generation of the RD signal which controls the output data. At any instant only one input stream is given to the output side.

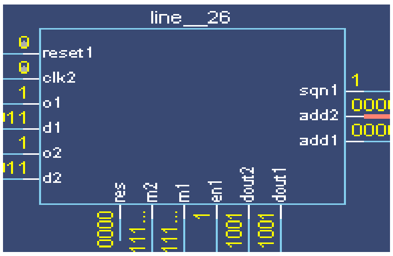


Fig. 13 Controller output to Buffer

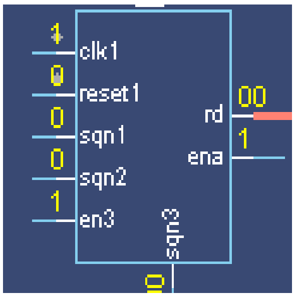
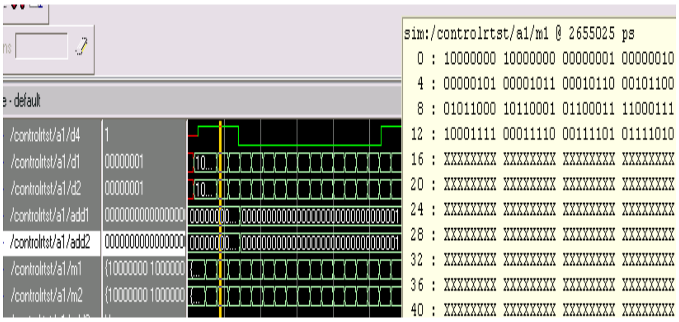


Figure 14 controller output to Data Generator


- default
/controlrtst/a1/d4 1
/controlrtst/a1/d1 00000001
/controlrtst/a1/d2 00000001
/controlrtst/a1/add1 0000000000000000
/controlrtst/a1/add2 0000000000000000
/controlrtst/a1/m1 {10000000 1000000
/controlrtst/a1/m2 {10000000 1000000
10...
10...
0000000...
00000000000000000000000000000001
0000000...
00000000000000000000000000000001

```
sim:/controlrtst/a1/m1 @ 2655025 ps
  0 : 10000000 10000000 00000001 00000010
  4 : 00000101 00001011 00010110 00101100
  8 : 01011000 10110001 01100011 11000111
 12 : 10001111 00011110 00111101 01111010
 16 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 20 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 24 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 28 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 32 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 36 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
 40 : XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
```

Figure 15 controller block waveform generating add signal

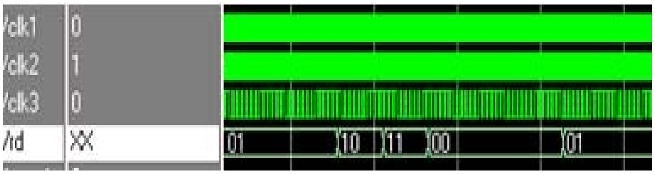


Figure 16 controller block generating RD signal

Fig 17 shows the block diagram of data generator block, which stores the synchronized input streams in m1, m2, m3 and m4 and then generates DOUT based on the value of RD signal.

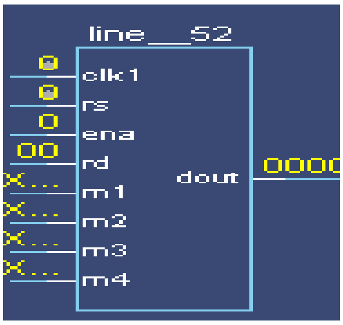
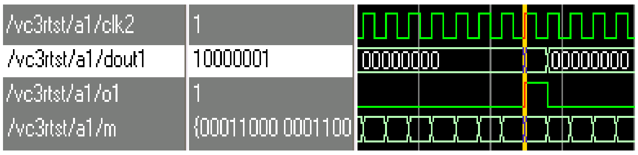


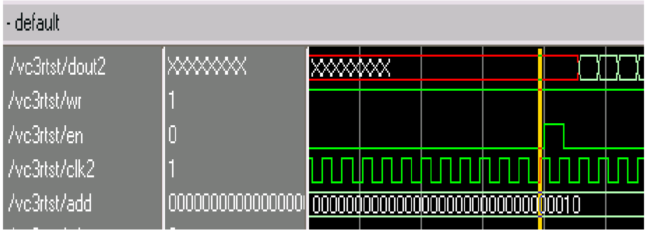
Figure 17 Data Generator Block

Fig 18 and 19 gives the waveform of VC-3 buffer block. Fig 18 shows the extraction of frame count and sequence number, which is then applied to controller block. It also gives O1 signal indicating the presence of H4 byte (frame count) on DOUT1 line. Fig 19 shows the output on DOUT2 given to data generator block when it receives EN and ADD signal from controller block.



```
/vc3rtst/a1/clk2    1
/vc3rtst/a1/dout1   10000001             00000000    00000000
/vc3rtst/a1/o1      1
/vc3rtst/a1/m       {00011000 0001100
```

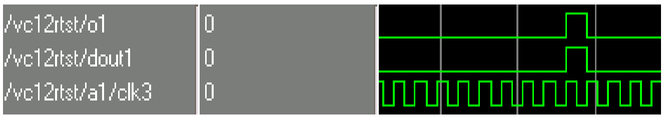
Figure 18 VC-3 buffer waveform output H4 byte for controller



```
- default
/vc3rtst/dout2   XXXXXXX            XXXXXXX
/vc3rtst/wr      1
/vc3rtst/en      0
/vc3rtst/clk2    1
/vc3rtst/add     0000000000000000   00000000000000000000000000000010
```

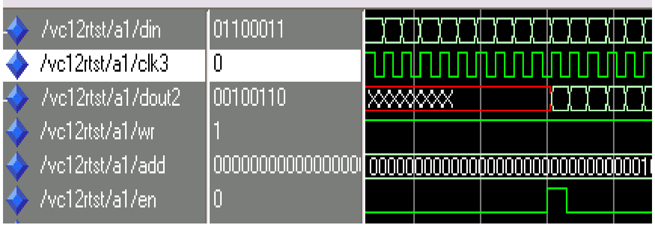
Figure 19 VC-3 buffer waveform output to the data generator block

Fig 20 and 21 shows the waveforms of VC-12 block. Fig 20 showing the 2nd bit of every K4 byte given to controller block on DOUT1 line and fig 21 shows the data given to data generator block on DOUT2 line just after it receives EN signal. Fig 22 shows the input and output waveforms of the complete receiver block.din1 to din4 are the four input streams and DOUT is the output 100 Mbps Ethernet data at clk1.



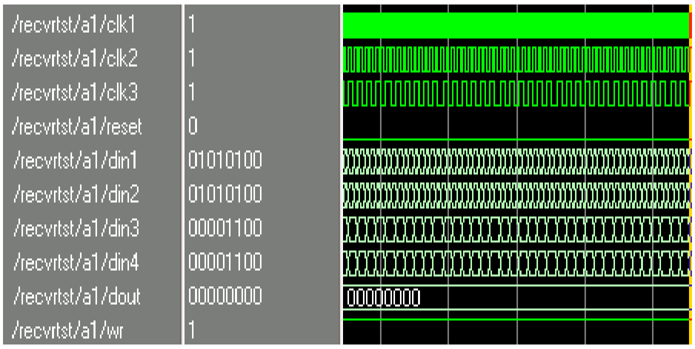
```
/vc12rtst/o1        0
/vc12rtst/dout1     0
/vc12rtst/a1/clk3   0
```

Figure 20 VC-12 buffer frame count to controller



```
/vc12rtst/a1/din    01100011
/vc12rtst/a1/clk3   0
/vc12rtst/a1/dout2  00100110           XXXXXXX
/vc12rtst/a1/wr     1
/vc12rtst/a1/add    0000000000000000   0000000000000000000000000000001
/vc12rtst/a1/en     0
```

Figure 21 VC-12 Buffer output to Data Generator



```
/recvrtst/a1/clk1    1
/recvrtst/a1/clk2    1
/recvrtst/a1/clk3    1
/recvrtst/a1/reset   0
/recvrtst/a1/din1    01010100
/recvrtst/a1/din2    01010100
/recvrtst/a1/din3    00001100
/recvrtst/a1/din4    00001100
/recvrtst/a1/dout    00000000   00000000
/recvrtst/a1/wr      1
```

Figure 22 Receiver input output waveform

V. Conclusion

CVC is going to replace the existing conventional VCAT technology as it has many advantages over it. CVC is more bandwidth efficient. It has also been proved that CVC provides greater flexibility in terms of path selection and reduces miss probability in connection establishment. CVC only needs hardware up gradation at the end nodes only, so the existing infrastructure can be utilized.

This paper describes the hardware implementation of CVC for 100Mbps Ethernet data transmission and reception using VHDL. It utilizes two VC-3 and two VC-12 channels for data transmission. The simulation results verify the functionality of circuit. The transmission delay calculated is 125us, which does not affect the designing or the functionality at the receiver as the whole stream is delayed by the same amount. Receiver circuit is designed for 32ms differential delay compensation which can be increased to 256ms by increasing the buffer size which is also calculated.

References

1. ITU-T Rec G.707; 'Network node interface for synchronous digital hierarchy'
2. ITU-T Rec G.7041; 'Generic Framing Procedure'
3. ITU-T Rec G.7042; 'Link Capacity Adjustment Scheme (LCAS) for Virtual Concatenated signals.
4. Satyajeet S. Ahuja and Marwan Krunz. "Cross-Virtual Concatenation for Ethernet-over-SONET/SDH Networks," Photonic Network Communications , Springerlink,April 2008
5. Canhui ou, Laxman h. Sahasrabuddhe, Keyao Zhu, Charles U Martel, Biswanath Mukherjee,'Survivable virtual concatenation for data over SONET/SDH in optical transport networks' IEEE/ACM transaction on networking, vol.14,no.1 Feb, 2006.
6. Hong Ju Kim, Seung Il Myong, Hyun Ha Hong, Jong Hyun Lee, JungSikKim, 'The Role of Ethernet over SDH in QoS Switch/router', IEEE ICACT, 2006.
7. Greg Bernstein, Diego Cleiglvia,Richard Rabbat, 'VCAT/LCAS in clamshell' IEEE comm. magazine, Vol.4 issue 5, May 2006, pp(34-36).
8. Han Dahai, Lili, 'Performance comparison of GFP and LAPS in application of data-transport', IEEE Communications, Circuits and Systems International Conference Proceedings, May 2005, pp(618-621).
9. Liangwei Ge,Takeshi Yoshimura, 'Design and implementation of an EOS chip' IEEE International Conference ASICON, Oct 2005, pp(300-303).
10. Xin Li, Depeng Jin, Lieguang Zeng, 'Encapsulation and rate adaptation for Ethernet over SDH', IEEE Communications, Circuits and Systems International Conference Proceedings, June 2002, pp(1301-1305).
11. Guowei shi,Qing wang, Zhaoliu, Lieguangg Zeng 'SDH Virtual concatenation technique used in ethernet data transport', IEEE Communications, Circuits and Systems International Conference Proceedings, June 2002, pp(1297-1300).