Parallel Positive Justification In SDH C_4 Mapping

Mao Shiwen Lin Xiaokang Shi Fuqiang

State Key Lab. of Microwave and Digital Comm.
Tsinghua University, Beijing 100084, China

ABSTRACT Bit Rate Justification is a key technique in digital multiplexing. SDH adopts Positive Justification for C_4 mapping and Positive/Zero/Negative Justification for other mappings. The mapping of C_4 has the highest system frequency in all Bit Rate Justifications of SDH. This demands highly on the technology and power consumption in the ASIC design. This paper puts forward a novel technique of Positive Justification with Parallel Processing and solves the problem caused by high speed. This method is very useful for implementing the C_4 mapping with CMOS Gate Array Technology. The design has been implemented with FPGAs of Xilinx Inc. The paper ends with the mapping jitter test results, which are quite satisfied. This paper is funded by the Project 863 of China.

Key Words: Synchronous Digital Hierarchy, Positive justification, Parallel processing

I INTRODUCTION

Bit Rate Justification is a key technique of Digital Multiplexing. In Plesiochronous Digital Hierarchy (PDH), the low-speed tributary signals are multiplexed into high level signal with positive Justification. The highest processing rate of Positive Justification in E_1 to E_4 is only about 34Mbps, which is easily implemented with CMOS ASIC technology. Since 1988, the Synchronous Digital Hierarchy (SDH) is recommended by ITU. The lowest bit rate of SDH is 155.520Mbps, which is called Synchronous Transport Module_1 (STM_1). For interface between SDH networks and PDH networks, a basic multiplexing structure has been recommended in ITU.T G.709. E_1 to E_4 data streams enter the SDH network via container mappings, which are Bit Rate Justifications. E_1 to E_3 signals use Positive/Zero/Negative Justification for mapping, while E_4 uses Positive Justification for Mapping, which is called C_4 mapping (Container_4 mapping). Among those mappings, the C_4 mapping has the highest processing rate, 140Mbps or 155.520Mbps, which brings difficulties to the implementation of ASIC design. This paper analyzes the difficulties caused by high processing rate and present a new solution to solve the problem.

The paper first describes the positive justification in SDH C_4 mapping. Then a conventional stuffing method which uses serial processing is given in part III. In part IV, a new solution of parallel synchronizer is put forward. Part V is the implementation and test results of the new solution.

II POSITIVE JUSTIFICATION IN C_4 MAPPING

At the boarder between SDH and PDH, signals from plesiochronous network must be mapped into correspondent SDH Container, and then be transported or cross-connected in the VCs within the SDH network. In fact, the SDH mapping process is Bit Rate Justification, which are very similar to the PDH multiplexing. The SDH mapping is a bit-oriented process. For example, the mapping of 139.264Mbps signal into C_4 is a Positive Justification with one justification opportunity bit each justification frame. Fig.1 shows one line of VC_4 frame structure [2].

As can be seen from Fig.1, there are three characteristics in the C_4 mapping. First, the number of non-information bits is rather large, which is due to the increased OAM bits and the fixed stuff bits. There are

$$(9+1+18)\times 8+1=225$$

non-information bits in one line of STM_1. This means that the coarse alignment is quite intense and that a quite large elastic buffer size is required in order to avoid an overflow of the buffer storage. Secondly, the system frequency is quite high. The write clock of the elastic buffer is 139.264MHz, and the read clock of the buffer is about 155.520MHz. So it is difficult to implement with common logical devices. If the elastic buffer is implemented with RAM, it will require highly on the write/read speed of the RAM; if the elastic buffer is implemented with flip-flops, the total circuit will be very complicated and very large. Since the system clock frequency is very high, the power consumption of the chip will be enormous when implemented with ASIC. The implementation is difficult and the cost is high. Thirdly, there are six information bits, one R bit and one S bit in the Z byte. S and R bit must be processed and the information structure is no longer byte-oriented.

0-7803-3925-8/97 $10.00 ©1997 IEEE

POH	W	96I	X	96 I	Y	96 I	Y	96 I	Y	96 I
	X	96 I	Y	96 I	Y	96 I	Y	96 I	X	96 I
	Y	96 I	Y	96 I	Y	96 I	X	96 I	Y	96 I
	Y	96 I	Y	96 I	X	96 I	Y	96 I	Z	96 I

W=IIIIIIII Y=RRRRRRRR X=CRRRRROO
Z=IIIIIISR I= Information bit R=Fixed stuff bit
O=Overhead bit S=Justification opportunity bit
C=Justification control bit

FIG.1 One Line Of VC_4

III COMMON SOLUTION

The conventional solution for such mapping uses serial processing. The block diagram of serial processing is given in Fig.2 [2].

Serial bit-shifted registers is often used as an elastic buffer. The data is written into the buffer storage with the clock CLK1 and read out of the buffer storage with the clock CLK2 bit by bit. Usually CLK1 is extracted from the input data stream while CLK2 is from the network element . The clock CLK2's frequency is a little bit higher than that of clock CLK1. The two counters CNT1 and CNT2 use CLK1 and CLK2 to generate write address and read address, respectively.

The clock alignment is accomplished in Fig.2 in two steps. The coarse alignment are usually done by insertion of fixed stuff or overhead information bits by use of the shown MUX block. For this purpose some gaps are inserted in the central clock by frame logic block.

The fine alignment is done by a locked loop consisted of CNT1, CNT2, COMP, CTR and gap machine. To prevent an overrun of the buffer storage due to the difference between the normal frequencies of CLK1 and CLK2, the write address and the read address are compared in the COMP block. The result of comparison, the address difference, controls the control logic CTRL and finally controls the amount of the gaps inserted in the

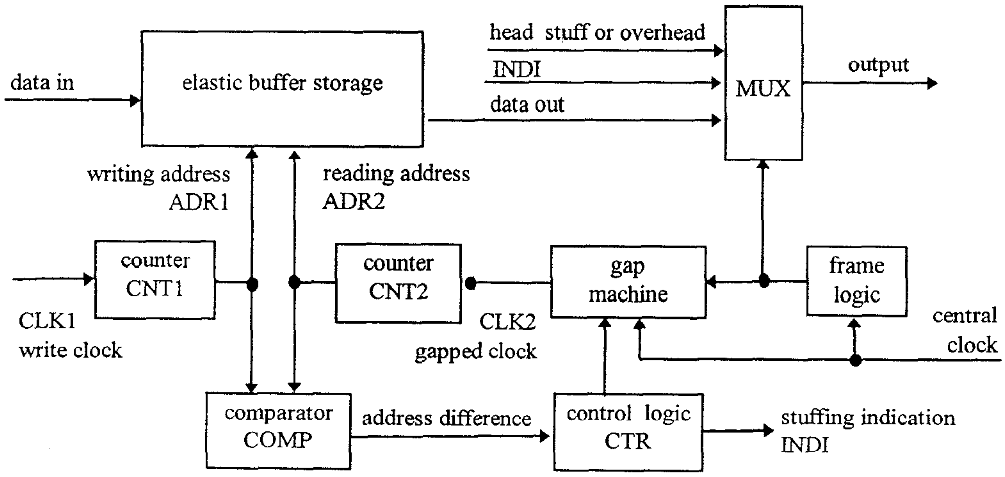

Fig.2 Block diagram of serial synchronizer

central clock. Every time when the read address reaches the write address, a gap of one bit is inserted in S bit position. Within a gap no data is read out of the buffer storage. This can ensure that in any time interval the same number of bits are written into the buffer as is read out of it. This also means that the data is clock aligned to the system clock, but with gaps. The stuffing indication bits INDI is multiplexed into the outgoing data stream at fixed positions in the frame of the signal and transported to the receiver. Using an elastic buffer and a Phase Locked Loop, the receiver can resume the original clock with the stuffing indication and the network clock.

In serial synchronizer, processing (e.g. storage and gap insertion) is bit by bit. So most of the circuits work at very high rate of 140MHz or 155MHz. It demands highly on the technology and power consumption, which also limits the chip capacity.

IV A NEW SOLUTION OF PARALLEL SYNCHRONIZER

A new technique of Positive Justification with Parallel Processing is put forward in this section. With the new technique, the circuit's working frequency is greatly reduced. Fig.3 shows the block diagram of synchronizing process.

The serial input data is first converted into parallel data, then is written into a buffer storage with the clock CLK1 and read out of the buffer storage with the clock CLK2. The coarse alignment is the same as what is shown in Fig.2. However , the fine alignment is quite different from the technique shown in Fig.2. The fine alignment is also done by a locked loop, but it is done with the CLK1. In this technique, CLK2 is a gapped clock obtained by insertion of gaps in the CLK3 while CLK3 is 1/8 of the central clock. CLK1 is obtained by processing tributary clock with a changeable counter CNT3 which is controlled by the stuffing request. When there is no requirement for stuffing and the current written byte is not Z byte, CLK1 is 1/8 of the tributary clock; when there is no requirement for stuffing and the current written byte is Z byte, CLK1 is 1/7 of the tributary clock; when Z byte comes and there is a stuffing requirement, CLK1 is 1/6 of the tributary clock. That is, the fine alignment is done with CLK1. CLK1 controls the serial to parallel converter CNV1. When it is time for writing Z byte into the buffer storage, the bit shifter in CNV1 is read out only 6 or 7 bit of information once, and this is controlled by whether this frame needs a stuff.

With this technique, the majority of the circuit work at about 20MHz while CNV1, CNV2, CNT3 and CNT4 work at 140MHz or 155MHz. The system working frequency is greatly reduced. Because the address difference is the interval between read address and write address, there is no difference between delaying the read clock of the buffer and accelerating the write clock. The gap machine is working at a kind of serial manner, so it is the same as serial processing when analyzing jitter. Those techniques for jitter reduction are feasible for this solution as well.

This technique has been applied for patent.

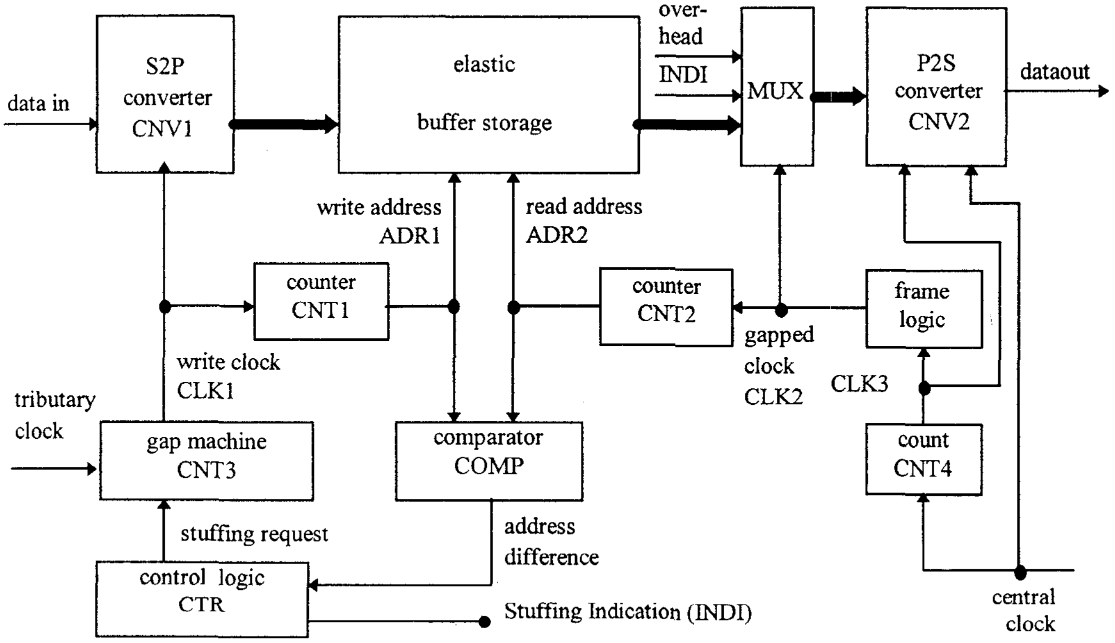

Fig.3 Block Diagram Of Parallel Synchronizer

Tab.1 shows the comparison between common serial mapping and parallel mapping. The differences lie in the gate counts of the circuit, the working frequency and the power consumption. Because the system working frequency is quite low with parallel processing, Dual-port RAM can be adopted as elastic buffer when implemented with ASIC; while in the case of serial processing, buffer cell can only be implemented with Flip-flop because of the technical limits. In contrast to serial processing, parallel processing not only reduce the requirements on the ASIC technique, but also lower the power consumption greatly, which is very meaningful to the ASIC design.

V IMPLEMENTATION AND TEST

For the verification of this parallel processing technique, a stuffing control logic of C_4 mapping was implemented with the FPGAs of XILINX Inc. The diagram of our experimental board is showed in Fig.4.

The majority circuit is held in the FPGAs. Two XC4005 chips were used, one for the synchronizer and the other for the desynchronizer. Circuits in the dashed box work at about 20MHz; while the circuits outside the box work at about 140MHz or 155MHz respectively. High speed logic, including CNT3, CNT4, CNV1 and CNV2 are implemented with high-speed GALs and ECL devices. The gate count for the circuits working at 20MHz is nearly 4000 gates, and the gate count for those circuits working at 140MHz or 155MHz is about 400 gates. There is an analogue phase locked loop in the desynchronizer, the PLL's cutoff frequency is 70Hz.

Mapping jitter is measured by transmission analyzer ME520B. The maximum peak-peak jitter under f1-f4 band-pass filter (200Hz to 3500kHz) is 0.10UI, and the maximum peak-peak jitter when there is no filter is 0.30UI. The detailed result is shown in Fig.5. The horizontal axis is the tributary clock offset and the unit is ppm; the vertical axis is the jitter amplitude measured and the unit is UI.

VI CONCLUSIONS

The mapping of E_4 signal into C_4 has the highest system frequency and the largest gate count of circuit among SDH mappings. It is difficult to implemented with serial processing. This paper presents a new solution of parallel processing, in which the system working frequency is only 1/8 of the former and the total power consumption is only 12.7% of serial processing. Both the system working frequency and power consumption are greatly reduced. Parallel processing is the much better solution for C_4 mapping, and it is very meaningful for ASIC design of mapping.

TAB 1. Comparison Between Serial Processing And Parallel Processing

	Gate count	Working frequency (MHz)	Power consumption (W)
Serial processing	4500	139,155	630000p
Parallel processing	4000	20	80000p

Note: p is the power consumption of ASIC chip per gate per MHz.

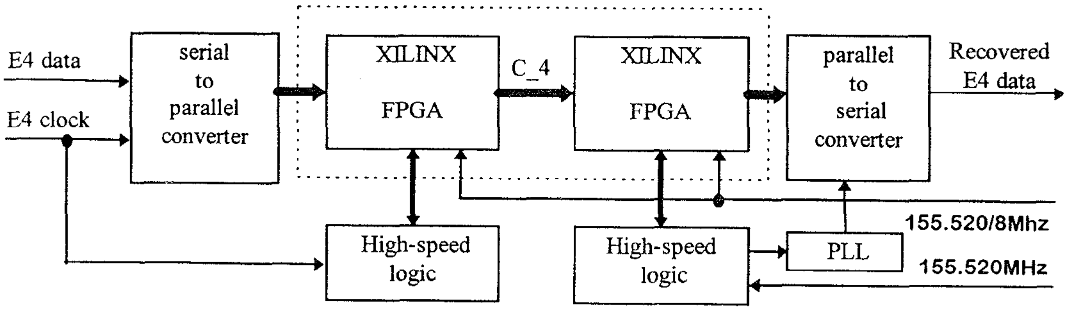

Fig.4 Diagram Of Experimental System

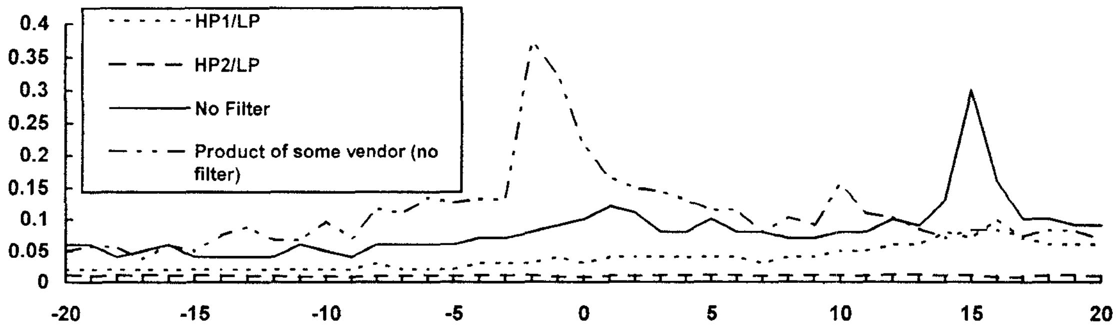

Fig.5 Jitter Test Result Of C_4 Mapping (UI p-p)

REFERENCES

[1] Sun yu, "Digital Multiplex Techniques", 2nd Edition, People's Posts and Telecommunications Publication House, Beijing, 1991.

[2] "NETWORK NODE INTERFACE FOR THE SYNCHRONOUS DIGITAL HIERARCHY (SDH)", ITU-T Recommendation G.709 (DRAFT).

[3] Pirmin-Nikolaus Foerg, "IMPROVEMENT OF THE JITTER BEHAVIOUR OF SDH TRANSMISSION EQUIPMENT", ICCT'92, Beijing , Vol.1, pp.11.08.1-11.08.4, Sept. 16-18 1992.