Design of Fast Core-Node Processor for Packet-Forwarding without Header Modification in Optical Networks

W. M. Wong and K. J. Blow
Photonics Research Group, Aston University, Aston Triangle, Birmingham B4 7ET, United Kingdom.
Email: w.m.wong@aston.ac.uk

Abstract: We present a design of a fast all-optical core-node processor that performs packet-forwarding in optical networks without header-modification. The design is based on bit-serial architecture using TOADs as logic-gates that perform modulo-arithmetic to forward packets.
OCIS codes: (060.4250) Networks, (200.3760) Logic-based optical processing

1. Introduction

Multiprotocol-Lamba-Switching has emerged as an elegant solution to meet the bandwidth-management and service-requirements for next-generation IP-based backbone-networks. This label-swapping approach is a minor concern in electrical core-nodes but remains a challenge to future optical core-nodes because packet-header modification is necessary. Therefore, a packet-forwarding scheme without any label-swapping for optical-networks known as the *key-identification-scheme*(KIS) has been proposed [1]. In KIS, the path through the network could be chosen by simply computing a *label*(L) at the ingress node, which could then be used to make routing decisions. A modular-arithmetic is performed electronically at the edge-nodes to compute the route-specific label based on two arrays containing the *output-ports*(P) of the traversed nodes and the *node-specific-keys*(N) of the network. The packets are then routed through the network by performing the following operation at each core-node:

$$P(\textit{output-port}) = L(\textit{label}) \ \textbf{mod} \ N(\textit{node-specific-key}) \qquad (1)$$

This scheme is depicted in Fig-1. Previously, we presented an all-optical core-node processor that performs eqn.(1) for packet-forwarding without any header-modification [2]. The design was based on a bit-serial architecture using TOADs [3] that realize AND, NOT and XOR logic-functions. A similar module of the processor has already been demonstrated [4] but the most challenging part remains to be the one-bit-delay optical flip-flop that may require photonic-integrated-circuit technology. In the basic design, the processor performs subsequent subtractions until an overflow is detected. However, the processing-time becomes prohibitively high if the difference between *label* and *key* is large, a situation that arises in a densely connected network.

2. All-optical design

A new design is presented here that improves the processing-time significantly as the *label* >> *key*. The overall architecture of the modulo-processor and its new modules are shown in Figs-2 and 3. The new design employs a shift-register to perform binary-multiplication before each subtraction. It also has comparator-logic-module that makes a decision whether to allow a subtraction. This method resembles the commonly known long integer-division method. This means that the processor could easily be modified into an optical-divider by keeping track of the number of subtractions performed. Previously, in the basic design, the total processing-time depends on the value of the *key* but here in the new design, the processing-time is fixed for a given m-bit *label*, which may be of another added advantage. The new design requires around the same number of gates compared to the previous design. We modeled the modulo-processor on a custom-written CAD-simulator based on an analytical TOAD model [5].

3. Results

Fig-4 shows the simulated bit-sequence of the fast-processor, where some amplitude-modulation is observed in (b) but the *thresholding* function of the gates suppresses this effect. Fig-5 shows the improved processing speed of the new modulo-processor design as compared to the basic design. The processing task scales up exponentially for the basic-processor while it is only quadratic for the fast-processor.

4. References

[1] H. Wessing, et al, IEEE J. Lightwave Tech., to be published.
[2] W.M. Wong and K.J. Blow, Proceedings COIN / PS 2002, pp.31-33.
[3] J.P. Sokoloff, et al, IEEE Photon. Tech. Lett., vol.5, pp.787-790, 1993.
[4] A.J. Poustie, et al, Opt. Comm., vol.168, pp.89-93, 1999.
[5] K.J. Blow, et al, Opt. Comm., vol.148, pp.31-35, 1998.

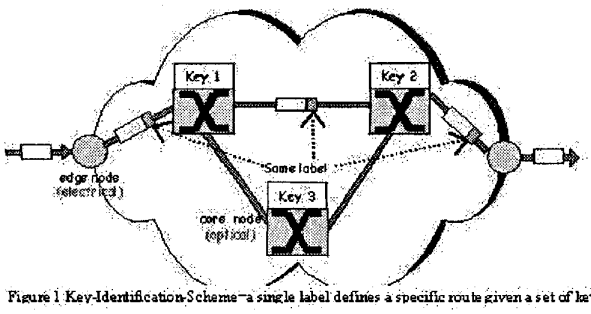


Figure 1 Key-Identification-Scheme—a single label defines a specific route given a set of keys in the network.

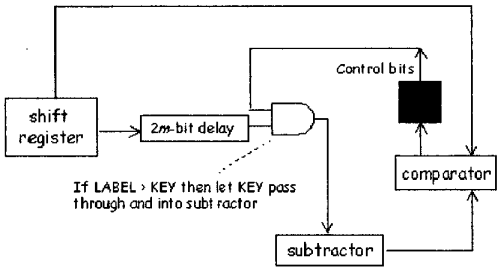


Figure 2 Overall modulo-processor architecture—the black box depicts different choice of implementation for deriving the control bits.

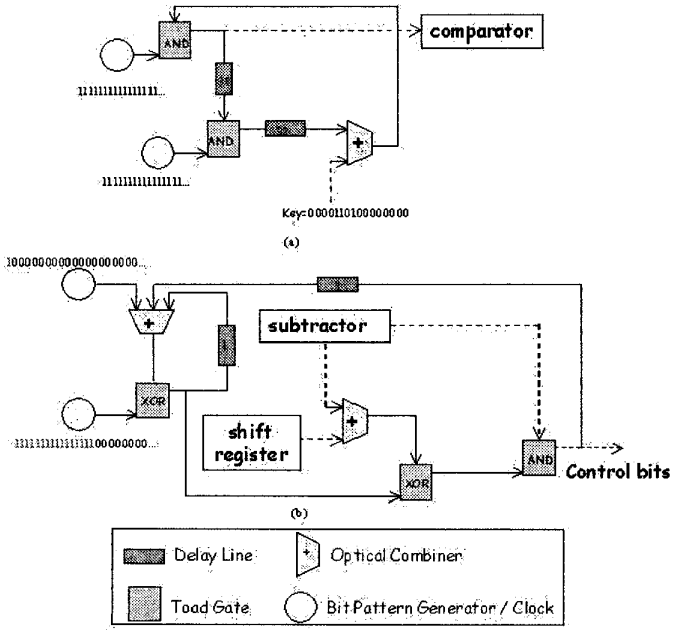
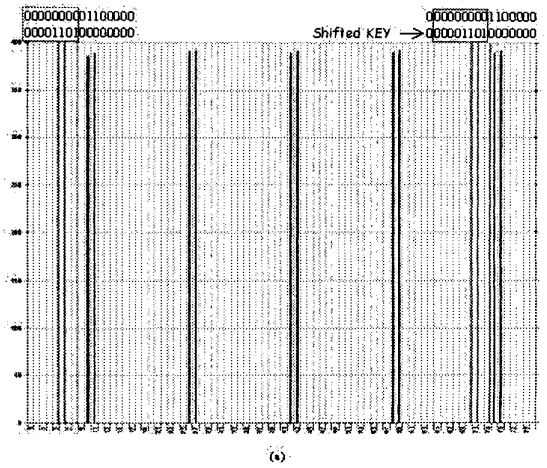


Figure 3 (a) Shift register module (b) comparator module.



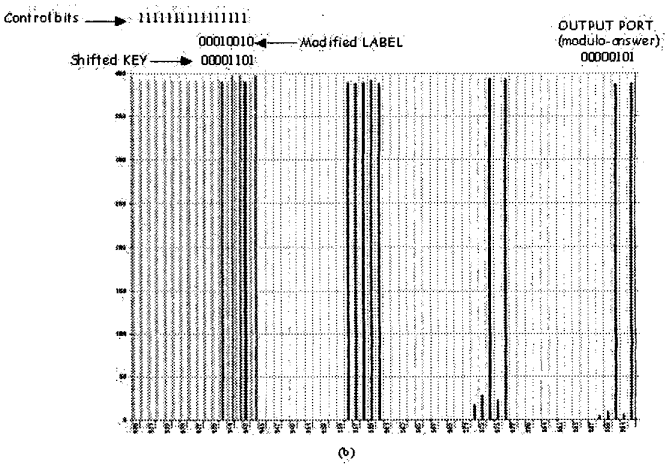


Figure 4 Simulated bit sequence of modulo-processor: (a) at comparator module (b) at input of subtractor module. The unshifted KEY is 13$_{10}$(00001101$_2$), the unmodified LABEL is 96$_{10}$(01100000$_2$). Bit no. 529-544 shows the alignment of shifted KEY with modified LABEL at the beginning of the final subtraction cycle. Lightly shaded lines are the control bits to 'allow' a subtraction to be carried out. The final modulo-answer is 5_{10}(00000101$_2$), which is the output port.

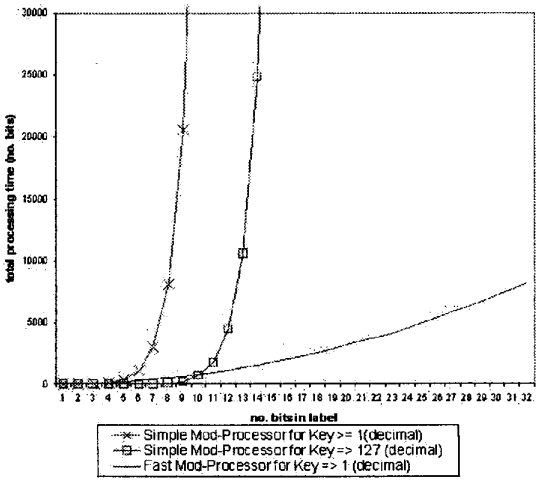


Figure 5 Scalability of (i) basic modulo-processor (ii) fast modulo-processor as m-bit label increases.