Improved Design of All-Optical Processor for Modular Arithmetic

W. M. Wong, K. J. Blow

Abstract
A new improved design of an all-optical processor that performs modular arithmetic is presented. The modulo-processor is based on all-optical circuit of interconnected semiconductor optical amplifier logic gates. The design allows processing times of less than 1μs for 16-bit operation at 10Gb/s and up to 32-bit operation at 100Gb/s.

1. Introduction

All-optical digital processing is an exciting area of research towards finding new ways to both process and transmit information entirely in the optical domain [1,2]. Owing to the fact that there is no equivalent of all-optical static memory, 'time-of-flight' processing based on bit-serial architecture has been proposed [3,4]. In recent years, there have been reports of all-optical logic gates [5-8,22-24] and all-optical processing circuits [9-11].

The basic design of an all-optical modulo-processor was first presented and described in detail in [12]. It is based on simple iterative subtraction to compute the modular arithmetic, $B = A \bmod N$. The basic design is limited to simple cases where the number of subtractions is small. In this letter, we present an improved design of the modulo-processor, which can be faster than the basic design by several orders. The modulo-processor may have potential applications in all-optical packet header processing [21] where routing information can be decoded from a field in the packet header using modulo arithmetic.

2. All-Optical Logic Design

The fast design is simulated by our all-optical circuit simulator [12]. The main component in the simulator is the all-optical logic gate based on the TOAD (Terahertz Optical Asymmetric Demultiplexer) configuration [13]. A block diagram showing the main modules of the fast all-optical modulo-processor is given in Fig. 1. The process mimics the simple method for doing integer division by hand. The denominator (strictly speaking, multiplied by powers of the number base) is subtracted from the most significant digit of the numerator and then shifted to less significant digits as appropriate. Since the principle behind the fast design is based on long-integer division in order to compute the modulo-answer, the fast modulo-processor is also an all-optical *divider*. The logical functions required in long-integer division and their corresponding modules are shown in Fig. 2. The example used here is 96 mod 13. The three main operations involved are (i) shifting the divisor (SHIFT DIV), (ii) comparing the input with the shifted divisor (COMPARATOR), and (iii) subtraction (SUBTRACTOR). The subtractor module is described in detail in [12]. The two new modules here are the SHIFT DIV and COMPARATOR modules. It is worth pointing out that the modulo-processor could be easily modified into an all-optical divider by simply keeping track of the number of subtractions in binary format (111_2 in the example).

For m-bit computation (m = 8 in the example), we require that the input A be $2m$-bit long with the first m significant bits padded with '0's. The SHIFT DIV module generates the divisor N when the 'sync' pulse arrives [12]. The divisor N is cyclically delayed by one-bit after every $2m^2$ bits. The COMPARATOR checks if subtraction should be performed. If A becomes greater than the shifted divisor N, a gating window (DIV CTRL) switches the shifted N into the SUBTRACTOR module. We should allow up to $2m^2$ bit-duration to ensure that subtraction is complete before the next shifted N arrives. To save computation time, we have used (m^2 = 64) bit cycles instead, which is valid in our example since all subtractions were found to be complete within this duration. After the divisor N has been shifted m times, then the modulo-computation is complete (see Fig.2).

The logical functions are realized by TOAD gates [13]. The TOAD gate in our simulator is based on a computationally efficient model that includes longitudinal effects [14] and is denoted by a four-port symbol here (Fig. 3).

The main application of the COMPARATOR module is to compare two bit streams and activates the SUBTRACTOR module if the logical condition is met where A (SUBTRACTOR output) is greater than N (SHIFT DIV output). In the COMPARATOR module (Fig. 4), the only time a logical 'one' is generated at the output AND gate (connected to the XOR gate) is when $A > N$. This happens when a 'one' is detected from the SUBTRACTOR module(1) and a 'zero' is detected from the SHIFT DIV module(0). This logical 'one' output from COMPARATOR will activate the DIV CTRL module which in turn generates a gating window to allow further subtraction. In contrast, if a 'one' is detected from SHIFT DIV(1) while a 'zero' is detected from SUBTRACTOR(0), subtraction will not be performed because we know that $A < N$. Note that whether ($A < N$) or ($A > N$) is detected, the logical 'one' from the XOR gate output will disable the COMPARATOR module (i.e. stop performing the comparison functionality) until the next cycle. If two 'ones' or two 'zeros' are detected, then the COMPARATOR continues to scan for the logical condition ($A > N$).

In Fig. 5, the logical 'one' from the COMPARATOR then generates a gating window that varies from $2m$ to $4m$-bits long. The logical 'one' pulse varies in position depending on when the logical condition ($A > N$) becomes true. The $2m$ to $4m$-bit long gating window would ensure that the $2m$-bit divisor N is fed into the SUBTRACTOR.

The SHIFT DIV is simply an all-optical re-circulating shift register with a feedback length of $2m^2+1$. The SHIFT DIV module shifts the divisor N towards the least significant bit (LSB) every $2m^2$ bits. The division would be complete after m shifts of the $2m$-bit long divisor.

Table 1
Parameter Values of the Semiconductor Optical Amplifier (SOA)

Symbol	Parameters	Values
E_{data}	Data pulse energy	10fJ
E_{ctrl}	Control pulse energy	400fJ

I_b	Bias current	173mA
L	SOA length	1000μm
A	Effective cross-sectional area	$0.2\mu m^2$
Γ	Optical confinement factor	0.3
τ_e	SOA carrier lifetime	300ps
a	Gain cross-sectional area	2.5×10^{-20} m^2
N_{tr}	Transparency carrier density	1×10^{24} m^{-3}
n_{eh}	SOA nonlinearity	2×10^{-26} m^3
λ	Operating wavelength	1.55μm

The modulo-computation begins with the arrival of the input packet *A*, which generates a 'sync' pulse from a self-switching TOAD [12,16]. The divisor *N* and other optical clock sources (INIT, 2*m*-WIN, and 4*m*-WIN in Figs. 4 and 5) are derived from the 'sync' pulse, which require some passive delay-lines and regenerative memory modules [15]. In our simulator, the clock patterns are preset. The 2*m*-WIN(or 4*m*-WIN) clock source supplies the 2*m*-bit (or 4*m*-bit) window (a stream of 'ones', loosely corresponding to the rail voltage in an electronic circuit). The 2*m*-WIN (or 4*m*-WIN) clock pattern is a stream of 2*m* (or 4*m*) 'ones' repeated after every $2m^2$ bits. The INIT pattern is a single 'one' repeated after every $2m^2$ bits.

3. Results and Discussion

The logical design was tested and simulation results of the fast modulo-processor showed the correct modulo-computation. As the logic circuit consists of several TOAD gates, the highly-efficient longitudinal model simulates the pulse energies rather than computationally intensive time-domain pulses [25]. Fig. 6(a) shows the beginning sequence when the input packet *A* (labelled as *input A* 0000000001100000) arrives and is compared with the divisor *N* (labelled as *shifted N* 0000110100000000). After m^2 bits, the divisor is shifted by one bit towards the LSB and is compared again, as can be seen in the right side of Fig. 6a. Fig. 6(b) shows the ending sequence when the comparator detects that the current *intermediate answer* (00010010) is greater than the shifted divisor *N* (00001101)and generates a *gating window* (1111111111111111). The gating window allows the shifted *N* to be fed into the subtractor for the final subtraction. Notice that the amplitude modulation effects in the third round (00000101) are quickly restored by the next (final) round.

We can now compare the total processing time (T_{mod}) of the basic design in ref.[12] and the faster design here. The total processing time of the basic design is expressed as $(m+1)^2 \times int(A/N)$ bits where *int* denotes rounding off to the nearest (smaller) integer. The fast design has a total processing time of $2m^3$ bits. In the basic design, T_{mod} depends on the relative size between input *A* and divisor *N*. Therefore, T_{mod} of the basic design becomes prohibitively large when $A >> N$. On the other hand, the fast design has a fixed T_{mod} for a given *m*-bit operation. Fig. 7 shows that the maximum T_{mod} of the fast design can be several orders smaller than the basic design. The maximum T_{mod} is defined by taking $A = 2^m-1$. The fast design achieves T_{mod} = 0.66μs for 32-bit operation at 100Gb/s or 0.82μs for 16-bit operation at 10Gb/s. Although we have assumed operation speeds at 1Gb/s that corresponds to a carrier lifetime of 300ps, it is possible to achieve operating speeds above 10Gb/s and possibly up to 100Gb/s by using an optical holding beam to reduce the carrier lifetime of the

semiconductor optical amplifier [20]. Table 2 lists some examples of T_{mod} values of the basic and fast designs with *A*, *N* and operating speed as parameters.

The proof-of-principle of the improved design is presented here by way of theory and simulations. In practice, the advanced all-optical logical design would be realised as fully integrated circuits [19]. In such cases, the transmission time is negligible compared to processing time. For example, if we have logical design consisting of 10 gates, the overall length would be about 10x1000um or 1 x 10^{-2}m. It would take approximately 50ps of transit time compared to the order of us of processing time. Also note that in the bit serial architecture internal gate delays merely add a small constant to the total processing time. Since the packet needs to be delayed, typically using a passive delay line, until the switch decision is available the internal delays can be easily accommodated.

Our work focuses on the logical design of the all-optical modulo-processor. There are additional problems to consider in its practical implementation [12]. Cross gain modulation (XGM) has been neglected in these simulations. However, by using a gain-transparent TOAD [17] and amplifier optimization [12], XGM effects can be minimized. In fact, the cascade feedback configuration of multiple TOADs provides amplitude restoration effects [18] as shown in Fig. 6(b). There are three tiny pulses which are quickly suppressed by the next round. In the practical design, it may be difficult to realize the single-bit delays by fiber delay-lines but monolithic integration technology will solve this problem [19].

Table 2
Processing Times of Modular Arithmetic Examples

Operation @10Gb/s	Basic Design	Improved/Fast Design
16383 mod 16	23.0μs	0.439μs
16383 mod 128	2.86μs	0.439μs
131071 mod 128	33.1μs	0.819μs
131071 mod 128 @100Gb/s	3.31μs	81.9ns

4. Conclusion

We have presented an improved design of an all-optical processor that performs modular arithmetic. The design is based on a bit-serial architecture with semiconductor optical amplifier-based logic gates as main building blocks. Simulations were performed demonstrating the correct logical operation of the processor. Our design requires 9 optical gates, such a low count is a result of the gate re-use inherent to the bit serial architecture. The modulo-processor has potential application in all-optical packet header processing.

5. References

1. R. J. Manning, A. D. Ellis, A. J. Poustie, K. J. Blow, JOSA B 14 (1997) 3204.

2. D. Cotter, R. J. Manning, K. J. Blow, A. D. Ellis, A. E. Kelley, D. Nesset, I. D. Phillips, A. J. Poustie, D. C. Rogers, Science 286 (1999) 1523.
3. H. F. Jordan, V. P. Heuring, R. Feuerstein, Proc. IEEE 82 (1994) 1678.
4. V. P. Heuring, H. F. Jordan, J. P. Pratt, Appl. Opt. 31 (1992) 3213.
5. T. Houbavlis, K. Zoiros, K. Vlachos, T. Papakyriakopoulos, H. Avramopoulos, F. Girardin, G. Guekos, R. Dall'Ara, S. Hansmann, H. Burkhard, IEEE Photon. Tech. Lett., 11 (1999) 334.
6. H. Soto, D. Erasme, G. Guekos, IEEE Photon. Tech. Lett., 13 (2001) 335.
7. C. Bintjas, M. Kalyvas, G. Theophilopoulos, T. Stathopoulos, H. Avramopoulos, L. Occhi, L. Schares, G. Guekos, S. Hansmann, R. Dall'Ara, IEEE Photon. Tech. Lett. 12 (2000) 834.
8. J. H. Kim, Y. M. Jhon, Y. T. Byun, S. Lee, D. H. Woo, S. H. Kim, IEEE Photon. Tech. Lett. 14 (2002) 1436.
9. A. J. Poustie, K. J. Blow, A. E. Kelly, R. J. Manning, Opt. Comm. 168 (1999) 89.
10. A. J. Poustie, K. J. Blow, A. E. Kelly, R. J. Manning, Opt. Comm. 162 (1999) 37.
11. A. J. Poustie, R. J. Manning, A. E. Kelly, K. J. Blow, Optics Express 6 (2000) 69.
12. W. M. Wong, K. J. Blow, Opt. Comm 265 (2006) 425.
13. J. P. Sokoloff, P. R. Prucnal, I. Glesk, M. Kane, IEEE Photon. Tech. Lett. 5(1993) 787.
14. K. J. Blow, R. J. Manning, A. J. Poustie, Opt. Comm. 148 (1998) 31.
15. A. J. Poustie, A. E. Kelly, R. J. Manning, K. J. Blow, Opt. Comm. 154 (1998) 277.
16. H. J. Lee, H. G. Kim, J. Y. Choi, K. Kim, J. Lee, IEEE Photon. Lett. 11(1999) 1310.
17. S. Diez, R. Ludwig, H. G. Weber, IEEE Photon. Tech. Lett. 11 (2000) 60.
18. A. J. Poustie, K. J. Blow, R. J. Manning, Opt. Comm. 146 (1998) 262.
19. V. M. Menon, W. Tong, C. Li, F. Xia, I. Glesk, P. R. Prucnal, S. R. Forrest, IEEE Photon. Lett. 15 (2003) 254.
20. R. J. Manning, D. A. O. Davies, Optics Lett. 19 (1994) 889.
21. H. Wessing, H. Christiansen, T. Fjelde, L. Dittmann, IEEE J. Lightwave Tech., 20 (2002) 1277.
22. J. Wang, J. Sun, Q. Sun, IEEE Photon. Tech. Lett. 19, (2007) 541.
23. J. Zhang, J. Wu, C. Feng, K. Xu, J. Lin, IEEE Photon. Techn. Lett. 19 (2007) 33.
24. J. Wang, J. Sun, Q. Sun, Optics Express 15 (2007) 1690.
25. W. M. Wong K. J. Blow, Optics Comm. 215 (2003) 169.