ASSPA 89, signal processing, theories, implementations and applications.
Editor: R.F. Barrett, Adelaide, Australia
17-19 April 1989

Systolic Convolvers in Parallel to Achieve Higher Throughput

H.A. COHEN
Computer Science Dept, La Trobe University, Bundoora, Vic.

SUMMARY: This paper describes a novel scheme for increasing the throughput of systolic convolvers. The same scheme can be applied to systolic pattern matchers.

Applied to convolution, the scheme involves a particular pattern of splitting the input stream to yield streams of sufficiently long unit duration to be input into a W1 convolver chain, the utilisation of a number of W1 chain convolvers in parallel, and a system of staggered merging via addition of the outputs of the parallel convolvers. The speed-up is linear in the number of W1 type convolving elements utilised but for each speed doubling further splitting and merging circuitry is required.

The scheme introduced here is especially attractive for use in conjunction with bit-serial and digit-serial silicon compilers.

1 Introduction

Convolution is a basic operation in signal processing. In this paper we use the notation,

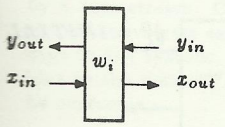
$$y_i = \sum_{j=0}^{N-1} w_j x_{i+j} \qquad (1)$$

to describe the convolution of an input stream x_i with convolution weights w_i for $i = 1 \ldots N-1$.

By introducing a variable $y_{i,j}$ where j ranges from -1 to N, one can recast the convolution summation as

$$y_{i,-1} = 0 \qquad (2)$$
$$y_{i,j} = y_{i,j-1} + w_i x_{i+j} \qquad (3)$$
$$y_i = y_{i,N-1} \qquad (4)$$

The feature of the above formulation is that a single multiplication and a single addition are performed at each stage of the iteration. This simplicity is a natural for simple hardware implementation as per the following unit, termed by Kung (1982, 1988) the W1 systolic convolver element.



$$y_{out} \leftarrow y_{in} + w_i x_{in}$$
$$x_{out} \leftarrow x_{in}$$

In each machine cycle of duration T the W1 element executes the algorithm presented above. Note that the two input operations take place at the beginning of the elements cycle, and the y_{out} can only be output after the arithmetic operations are completed.

Stringing a line of W1 elements together, one for each weight, yields a convolver as shown:

This is the simplest truly systolic convolver. It is most natural that input and output streams travel with the same speed, so

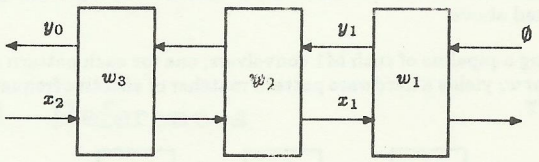


Figure 1: Systolic Array for performing three-fold 1-D convolution Classified by Kung as type W_1

that a "double-spacing" becomes necessary. As indicated in the figure, every second quantity in the input stream is one of the input $x_1, x_2, x_3, \ldots$ while the output stream, which traverses the one-dimensional systolic array in the opposite direction, likewise contains the desired $y_1, y_2, y_3, \ldots$ at every second quantity. Thus if the individual elements have machine cycle (period) time T, the throughput frequency is one half the element frequency, and is (1/2T).

In sum, this design may be characterised as only 50% efficient, having constant lag, and as having operational frequency 1/2T.

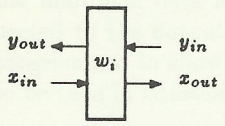
Cohen (1988) has pointed out that using dual weight convolving elements which use different weights in alternate cycles it is possible to design a convolver (termed by Cohen the WD) using an interleaved data stream that is 100% efficient, of constant lag, and with operational frequency 1/2T.

In this paper we further extend the scheme of stream splitting used in the WD convolver design (Cohen, 1988) to yield a new architecture for what is termed the P family of convolvers. In the Pn, where n=1,2, ... convolver, there are n convolvers of Kung's type W1 per weight. The throughput (= operating frequency) of the Pn convolver is simply 1/nT , where T is the machine cycle of the W1 elements. Thus there is a speed-up under this scheme linear in the number of W1 systolic elements. The overheads for beam splitting and merging are modest. Each Pn design is 100% efficient as there are no idle machine cycles for the convolving elements.

2 A Systolic Pattern Matcher

For this discussion, a systolic element - the M1 - is defined that is rather more analogous to the W1 convolving element than the more general element introduced by Foster and Kung (1980). The M1 element detects the presence in the input stream of

a fixed element (character etc), and appropriately ANDs the cumulative logical output stream element; this is strictly analogous to the W1 element where the next element in the input stream is multiplied by the (fixed) weight, and the result arithmetically added to the cumulative convolution output stream element. The same topology applied to the the W1 to produce speed-up through parallelism is relevent to the M1, as illustrated below:



$$y_{out} \leftarrow y_{in} \; AND \; (w_i = x_{in})$$
$$x_{out} \leftarrow x_{in}$$

In the expression above for y_{out} the MATCH function $(w_i = x_{in})$ returns a value 1 for a match, 0 otherwise. (If w_{in} is the 'don't care' pattern element a 1 is always returned). In each machine cycle of duration T the M1 element executes the algorithm presented above.

Using a pipeline of such M1 convolvers, one for each pattern element w_i yields a hardware pattern matcher of effective frequency 1/2T.

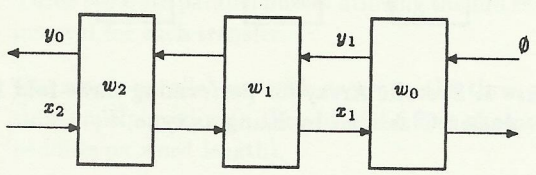


Figure 2: Systolic Array for performing pattern matching: using type M1 systolic elements

For this pattern matcher, with elements $w_0 = a$, $w_1 = b$, $w_2 = c$, and the input stream

xyzabcdexabc

the output is just

000100000100 .

It is stressed that this is only a limitted pattern matcher, matching a single pattern, but useful to show the relevance of the systolic approach and the parallelism scheme developed in this paper. o

3 The P1 Systolic Convolver

In this section we present a design for double stream systolic convolvers and pattern matchers that operate at 100% efficiency. This design is termed the P1.

For definitess, the discussion is couched in terms of convolution. Consider the expression for a 1D convolution:

$$y_i = \sum_{j=0}^{N-1} w_j x_{i+j} \qquad (5)$$

To simplify the following discussion we introduce (as necessary) the additional weights

$$w_i = 0 \quad i > N \qquad (6)$$

By making the separation

$$y_i = y_i^A + y_i^B \qquad (7)$$

into even and odd sums per the following, introducing E which for even N equals (N/2):

$$E = (N+1)div2 \qquad (8)$$

wherein

$$y_i^A = \sum_{e=0}^{E} w_{2e} x_{i+2e} \qquad (9)$$

$$y_i^B = \sum_{e=0}^{E} w_{1+2e} x_{i+2e} \qquad (10)$$

(11)

Implementation is quite direct, using the W1 systolic elements described above. Consider the design below, termed the P1, involving two parallel arrays of W1 elements, row A containing the odd weights, and row B contains the even weights. We suppose that the input stream comprises

$x_1, x_2, x_3, \ldots$ at unit time T spacing, and that this input stream enters BOTH rows, but that the stream is lagged by unit time T before entering row B. The outputs from both convolver rows are simply added.

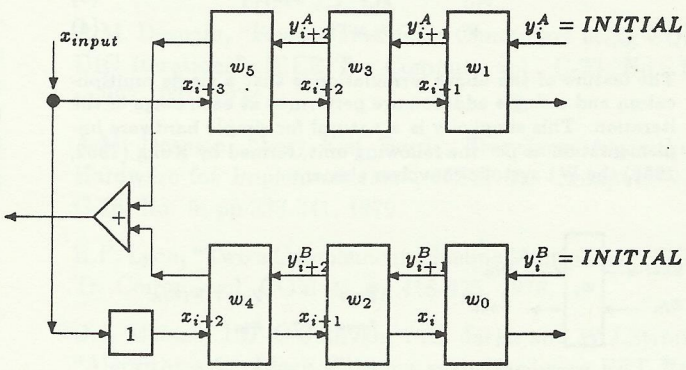


Figure 3: Systolic Array P1: where elements are W_1 type the array performs convolutions, using the weights $w_0, w_1, w_2, w_3, w_4, w_5$, while if M_1 are used this performs pattern matching against the pattern $w_0 w_1 w_2 w_3 w_4 w_5$

Note that there are no idle cycles for the P1, it functions with a frequency of (1/T).

4 The P2 Systolic Convolver

In order to determine the next member of this family, it is necessary to clarify the objectives. For an input stream with unit duration T, the P1 involved systolic elements of period 2T, and involved two parallel streams. For the P2, it is therefore specified that the systolic elements involved have period 2T. This entails that the stream through each systolic convolver row shall comprise elements x_i and x_{i-2} , and must involve a four-fold decomposition of the input stream like so:

$$y_i = y_i^A + y_i^B + y^C + y_i^D \quad (12)$$

where each term corresponds to a different partition of the convolution sum.

$$y_i^A = \sum_{f=0}^{F} w_{4f} x_{i+4f} \quad (13)$$

$$y_i^B = \sum_{f=0}^{F} w_{1+4f} x_{i+1+4f} \quad (14)$$

$$y_i^C = \sum_{f=0}^{F} w_{2+4f} x_{i+2+4f} \quad (15)$$

$$y_i^D = \sum_{f=0}^{F} w_{3+4f} x_{i+3+4f} \quad (16)$$

$$(17)$$

In the above, F is the smallest integer larger than (N/4):

$$F = (N+3) div 4 \quad (18)$$

Clearly a system following this algorithm, will compute in sequence y_i, y_{i+2}, ... So that there will need to be a double set of convolver arrays, as shown in Figure 4.

Exactly the same architecture applies to the systolic matcher based on P2, with M1 elements replacing W1 elements, and initial value of Y stream being 1 in lieu of 0.

5 Conclusions

The systolic concept of KUNG and co-workers (Kung 1988), is essentially a refinement of the pipeline concept, wherein processing elements operating in parallel are encountered sequentially by a data stream. On the other hand, the P1 and P2 parallel stream convolvers essentially adjoin the vector processing concept to the systolic. By adding the vector (array) processing idea to the systolic, some of the limitations of the systolic can be overcome.

With current fabrication technology, it is now relatively straightforward to design systolic elements using bit-serial silicon compilers such as FIRST. (Denyer, 1985). Cathedral (Rajeev et al 1986), BSSC (Yassa et al 1987). However convolvers designed in this manner suffer throughput limitations. For example, with 10Mhz clock, and 10 unit periods per data element, the pipeline unit time is just $10^{-}6$ sec. This is slower than for instance, the pixel duration in a video frame digitised at 512*512, which is $1.4 * 10^{-7}$. It is clear that utilising the speed-up with the P1 and P2, fuller utilisation can be made of bit-serial design.

Clearly parallel (rather than bit-serial) computation has the highest throughput - but at a high cost in silicon area. Rather than use only bit-serial multipliers, a compromise would be to use parallel multipliers of a digit size smaller than the word size. Thus Smith et al (1987) have suggested the use of radix-4 modes for higher performance in bit-serial computation. Hartley and Corbett (1988) have made a detailed analysis of efficiency (in terms of throughput/silicon area) and present as their conclusions that digit/serial is more efficient than word-parallel computation. They determined that for a class of fillter, digit-size in range 4-8 bit were optimum, and suggested that "the highest sample rates can be achieved by splitting the computation into parallel computational streams ... 4-8 bit [width]". It is clear that the parallel scheme of the P1 and P2 are further alternatives, whose precise efficiency, especially in conjunction with digit-serial compilation, remains to be determined.

6 References

COHEN, H.A. (1988). *Symmetry Considerations Applied to Hardware Convolvers for Image Filtering* IEEE SMC Conference, Beijing/Shenyang, August 1988, Vol 2, pp 1128-31.

DENYER, P. and RENSHAW, D. (1985). *VLSI Signal Processing: A Bit-Serial Approach* Addison-Wesley Publishing, Reading, Mass.

FORTES, J.A.B. and WAH, B.W. (1987) *Systolic Arrays - From Concept to Implementation* IEEE Computer Magazine Vol 20 No 7 July 1987 pp 12-17

FOSTER, M.J.(1980) and H.T. KUNG, *The Design of Special-Purpose VLSI Chips* Computer Vol 13 No 1 January 1980 pp 26-40

HARTLEY, R.I. and CORBETT, P. (1988). *A Digit-Serial Silicon Compiler* Proceedings 25th IEEE Design Automation Conference June 1988 pp 646-49.

JASICA, J.R. HARTLEY, R. et al (1985). *A Bit-serial Silican Compiler* Proc ICCAD-85 pp 91-93.

KUNG, H.T.(1982) *Why Systolic Architectures?* IEEE Computer Magazine January 1982 pp 37-46

KUNG, S.Y. (1988). *VLSI Array Processors* Prentice-Hall, Englewood Cliffs, New Jersey, 1988

SMITH, S.G. , McGREGOR, M.S. and DENYER, P.B. *Techniques to Increase the Computational Throughput of Bit-Serial Architectures* Proceedings of ICASSP'87, April 1987 pp 543.

YASSA, F.F. JASICA, J.R. HARTLEY R.I. and S.E. NOUJAIM, S.E. (1987). *A Silicon Compiler for Digital Signal Processing: Methodology, Implementation, and Applications* Proc IEEE Vol 75 No 9 pp 1272-1281

ZHOU, B.T. and BRENT, R.P. (1987) *An Efficient Architecture for Solving the Recursive Convolution Equation with High Throughput* Proceedings of the 1st IASTED International Symposium on Signal Processing and its Applications Brisbane, Australia, August 1987 pp 771-775

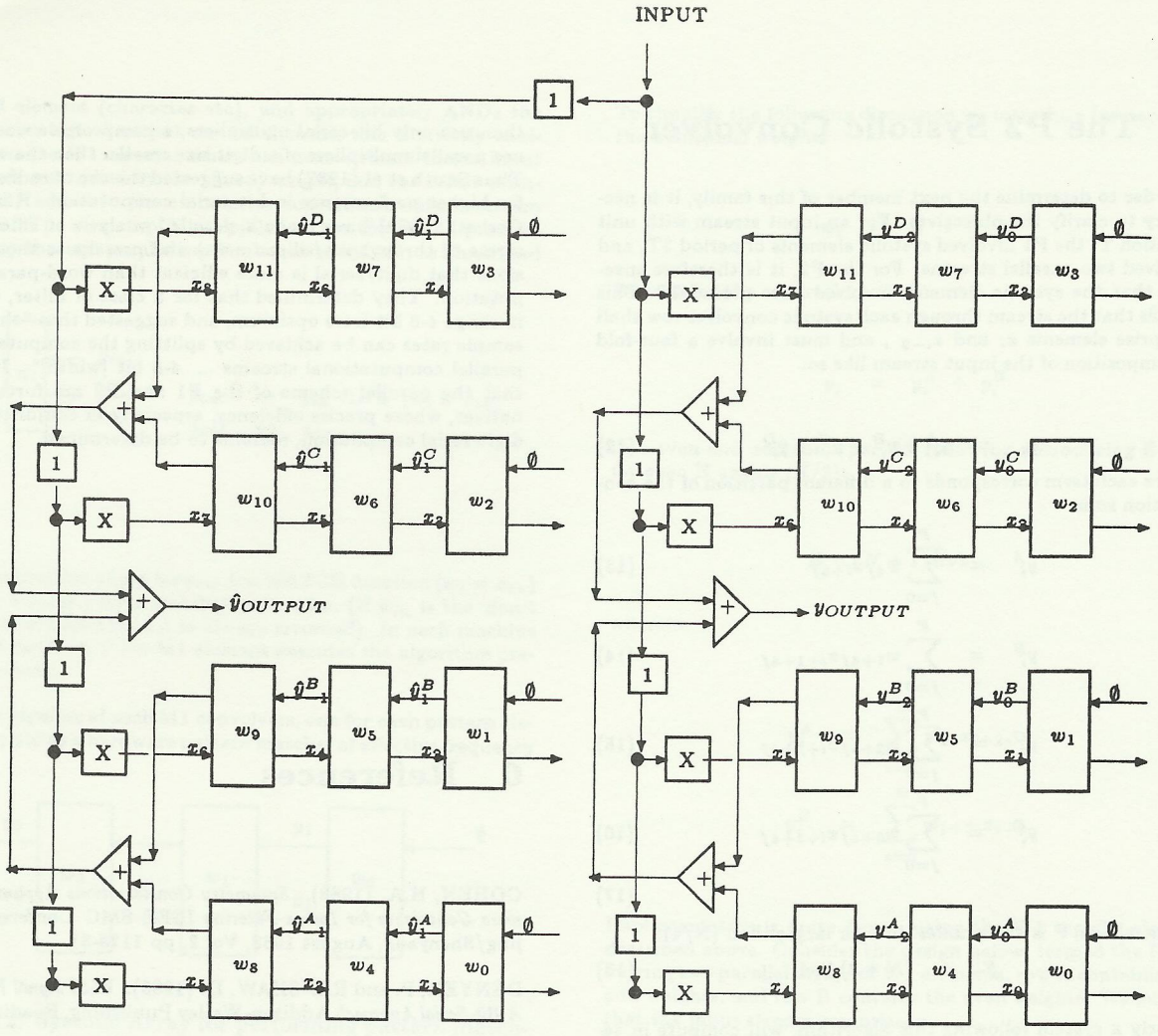


Figure 4: Systolic Array P2: This systolic convolver comprises two identical arrays, with the input to the left-hand array lagged one unit time with respect to the right-hand array. The P2 has two W1 elements per weight, and has a two-fold speed-up with respect to the P1. Note that the clocked buffers marked with an X permit only every second item to pass. The output of the left-hand array is just unit time lagged with respect to the output of the right-hand array. Note however that the period of the W1 elements is two unit times.