A New Parallel Algorithm for Lowpass Digital Image Filtering Using Systolic Array

Abdulraouf Y. Al-Hallaq
University of Jorndan
Saad A. Amin
Loughborough University

ملخص

يقدم هذا البحث خوارزمية جديدة تعمل على التوازي لتنقية الصور النقطية وزيادة وضوحها وذلك باستخدام مصفوفة المعالجات الانقباضية. هذه الخوارزمية تعطي نتيجة هذه العملية في زمن أقل من الخوارزميات التي سبقتها. ذلك أنه في السابق كانت عمليات الجمع في هذه المعالجات الانقباضية تتم بطريقة متتابعة، معالج انقباضي بعد الآخر حيث أن كل معالج ينتظر نتيجة الجمع من الذي يسبقه فيجمعه ويرسله للتالي.

أما في هذا البحث فإن المصفوفات الانقباضية ومن ثم عمليات الجمع ستقسم إلى مجموعتين، كل مجموعة تجمع نصف عمليات الجمع بطريقة متتابعة ولكن تعمل المجموعتان معاً وفي آن واحد وعلى التوازي وذلك بطريقة الطي. وبذلك يتم اختصار الزمن لهذه الخوارزمية الجديدة بواقع ربع الزمن المستغرق في الخوارزمية السابقة.

ABSTRACT:

This paper introduces a new parallel algorithm for the lowpass digital image filtering using a systolic array. This new algorithm is faster than the old one [14]. This is due to the fact that the old algorithm carries out the addition operations in a sequential mode. But in our new design these addition operations are divided into two groups which can be performed in parallel. One group will be performed on one half of the systolic array and the other on the second half, that is, by folding. This parallelism reduces the time required for the whole process by almost quarter the time of the old algorithm.

2. Introduction

Image processing is one of the areas in which most of the tasks are computationally intensive due to the large amount of data and the time-consuming repetitive operation of the algorithms in use. For example, a coloured image of the size 1024 requires 2M bytes, assuming each colour has 32 gray levels. The image goes through several levels of processing from the level of the numeric operations to the level of the symbolic operations. At each level the algorithm is time consuming. Therefore, the computational demand is far above the capacity of the conventional computers especially in the case of the real time applications where a sequence of images should be processed in a very short time.

To speedup the operation, several parallel architectures and processing methods have been discussed at various levels of the image processing [1]. They may be classified into special-purpose [2]-[4] or general-purpose [5], [6].

One of the tasks in image processing that may be processed in parallel is the image smoothing operation using a low-pass digital image filtering [7], [8]. It is possible to implement such a parallel algorithm on a printed board or a chip due to the dramatic development of VLSI technology.

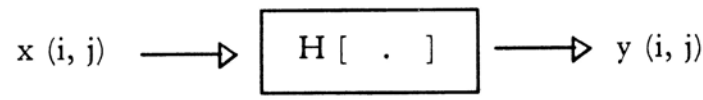
Following kung's systolic concept [9]-[11], many systolic arrays have been proposed to solve various problems [12], [13]. These array processors generally consist of a regular array of simple and identical processing elements (PE's) which rhythmically compute and pass data through the system. The systolic system provides a realistic model of computation which captures the concepts of pipelining, parallelism and interconnection structure.

The purpose of this paper is to introduce a new parallel algorithm using a systolic design for the low-pass digital image filtering which can be implemented as an Occam program on a Transputer Development System, and to compare it with the old design proposed in [14].

3. Digital Image Filtering And The Convolution:

Enhancement filtering of an image, low-pass for smoothing or high-pass for edge enhancement is an attempt to improve the quality of the image for human or machine interpretability, where quality is measured subjectively. Most of the enhancement filters are heuristic and problem oriented, [15], [18].

Mathematically, theory of linear systems is used to describe optical systems and provides the mathematical basis for certain filters used in digital image processing. Let x(i, j) and y(i, j) represent the input and the output sequences respectively, of a 2-dimensional system then the system



that can be written as

$$\mathbf{y\ (i,\ j) = H\ [x\ (i\ ,\ j)\ ]} \quad \text{..... (1)}$$

is linear if and only if any combination of two inputs x_1 (i, j) and x_2 (i, j) produces the same combination of their respective outputs y_1 (i, j) and y_2 (i, j), that is,

$$\mathbf{H\ [a_1x_1\ (i,\ j) + a_2x_2\ (i,\ j)] = a_1H\ [x_1\ (i,\ j)] + a_2H\ [x_2\ (i,\ j)]}$$

$$\mathbf{= a_1y_1\ (i,\ j) + a_2y_2\ (i,\ j)} \quad \text{..... (2)}$$

This is called a linear superposition, and when the input is the 2-dimensional kronecker delta function at the location (k,I) then the output at (i,,j) is defined as:

$$\mathbf{h\ (i,\ j;\ k,\ l) = H\ [\delta\ (i\text{-}k,\ j\text{-}l)]} \quad \text{..... (3)}$$

and is called the impulse response of the system which is, in imaging system the image in the output plane due to an ideal point source at (k,l) in the input plane. In our notation, the semicolon (;) is used to distinguish between the input and the output pairs of the coordinates.

The output of any linear system can be obtained from its impulse response together with its input by applying the above superposition rule to give

$$y(i , j) = H [x(i , j)]$$

$$= H [\Sigma_k \Sigma_l x(i , j) \ \delta(i - k ; j - l)] ,$$

and by using equ. (3), this give

$$y (i, j) = \Sigma_K \Sigma_l x (k, l) \ h(i,j; k, l) \quad \text{..... (4)}$$

Also a system is called spatially invariant or shift invariant if a translation of the input causes a translation of the output. Thus, following equ.. (3), if the impulse occurs at the origin then we will have

$$H [\delta (i, j)] = (i, j; 0 , 0)$$

Hence, it must be true for shift invariant systems that

$$h (i, j; k, l) = h (i - k ; j - l) \quad \text{..... (5)}$$

That is, impulse response is a function of the two displacements k, l. This means that the shape of the impulse does not change as the impulse moves about the i, j plane.

The result we want from system theory is that if a filter satisfies certain conditions, such as if it is linear and shift invariant, then the output of the filter can be expressed mathematically as:

$$g(i , j) = \Sigma_{i,j=-\alpha}^{\alpha} \Sigma h(i - k , j - l) \ f(k , l) \quad \text{..... (6)}$$

where h (k, l) is called the impulse response or the point spread function which completely characterises the filter.

The expression in equ.. (6) is a common form and is called a convolution and is written as:

$$g = f * h$$

Although the limits on the summation in equ. (6) are infinite, the function h may have value of zero outside some range. If h is non zero for $-w < k < w$ and $-v < l < v$ and zero else where, then equ. (6) can be written as:

$$y(i, j) = \sum_{k=i-w}^{i+w} \sum_{l=j-v}^{j+v} f(k, l)\, h(i - k, j - l) \qquad \text{..... (7)}$$

This says that the output y(i, j) at the point (i, j) is given by the weighted sum of the input pixels that surrounds i, where the weights are given by h(k,l), which is generated by a series of shift, multiply and sum operations. The values of h are also referred to as the filter weight, the filter kernel, or the filter mask.

4. Systolic Array:

Consider the problem of multiplying a matrix $A = (a_{i,j})$ with a vector $x = (x_1, x_2, \ldots, x_n)^T$. The resulting vector $y = (y_1, y_2, \ldots, y_n)^T$ can be computed by the following recurrence relations [11]:

$$y_1^1 = 0$$
$$y_i^{k+1} = y_i^k + a_{ik}\ x_k$$
$$y_i = y_i^{n+1}$$

For an nXn band matrix A with a bandwidth w = p + q-1. The above recurrences can be evaluated by pipelining the x_i and y_i through a systolic array consisting of w linearly connected inner product step processors. Each processor has three registers R_A, R_B, and R_C. In each unit time, the processor shifts the data on its input lines denoted by A, B and C into R_A, R_B and R_C, respectively, computes $R_C = R_C + R_A * R_B$, and makes the input values for R_A and R_B together with the new value of R_C available as outputs on the output lines denoted by A', V' and C', respectively, see Figure (1). All outputs are latched and the logic is clocked so that when one processor is connected to another, the changing output of one during a unit time interval

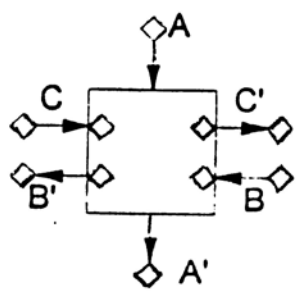
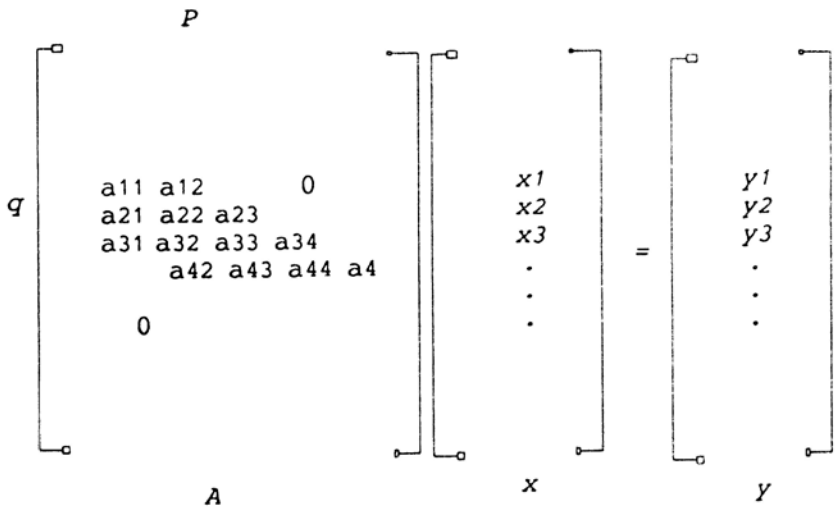
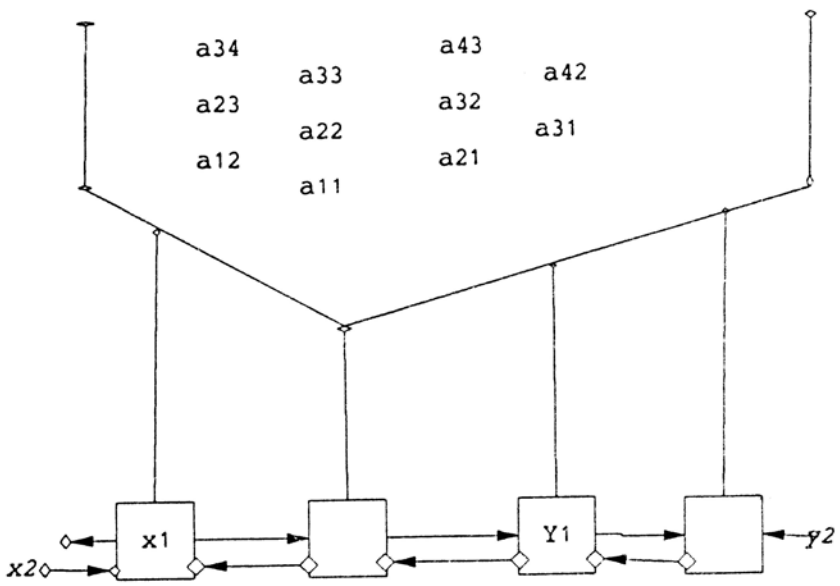


Figure (1)



(a) Multiplication of a vector by a band matrix , p=2,q=3



(b) Linear connected systolic array for (a).

Figure (2)

will not interfere with the input to another during this time interval. In the case of $p=2$ and $q=3$, four inner product step processors [11] are needed, see figure (2).

The computation as discussed in [11] proceeds as follows: The y_i's, which are initially zeros, are pumped from the right to the left while the x_i's are pumped from the opposite side from the left to the right and the a_{ij}'s are marching down. All the moves are synchronized. Each y_i is able to accumulate all of its terms, $a_{i,i-2} \cdot x_{i-2}$, $a_{i,i-1} \cdot x_{i-1}$, $a_{i,i} \cdot x_i$. x_i and $a_{i,i+1} \cdot x_{i=1}$ before it leaves the network.

Assume that the processors are numbered as 1,2,..., w from left to right and each processor has three registers, namely, R_A, R_x and R_y. All these registers are initialized to zero and will hold entries in A, x and y of figure (2b) respectively. And assume that for odd numbered pulses only even numbered processors are activated. Then each pulsation of the systolic array consists of the following operations:

1. Shift

- R_A gets a new element in the band matrix A.
- R_x gets the content of the R_x from the left neighbour (processor 1 gets a new component of x.)
- R_y gets the content of the R_y from the right neighbour (processor 1 outputs its R_y contents and the R_y in processor w gets zero.)

2. Multiply and add

- $R_y \leftarrow \ldots\ldots R_y + R_A \cdot R_x$

Using the inner product step processor, we note that the three shift operations in step 1 can be done simultaneously, and that each pulsation of the systolic array takes a unit of time. Suppose the bandwidth of A is $w = p + q - 1$. It turns out that after w units of time the components of the product $y = A_x$ are pumped out from the left processor at a rate of one output every two units of time. Therefore, in the systolic network all the n components of y can be computed in $2n + w$ time units, as compared to the $0(wn)$ time units needed for a sequential algorithm on a uniprocessor computer, for more details see [11].

5. Parallel Algorithm for Low-Pass Digital Image Filtering:

5.1 The Old Algorithm - The 2-D convolution:

From a computational point of view, an efficient way of computing a 2-D convolution is to transform it into that of computing a 1-D convolution. Therefore, the 2-D convolution can be performed on a systolic array for a 1-D convolution.

Consider an image matrix of size nxn and a kernel w of size mxm, figure (3) shows the image and the kernel for the case $n = 5$ and $m = 3$, then each row i of the image can be represented as:

$x_{i!}$,

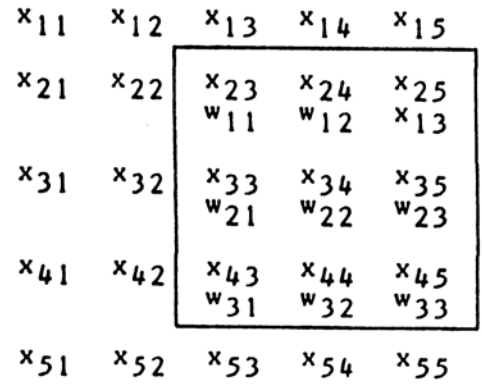
where $x_{i!} = x_{i1}, x_{i2}, \ldots, x_{in}, \quad i = 1 \ldots n$

and the image is represented as:

$x = x_{1!}, x_{2!}, \ldots, x_{n!}$

with a total length of n^2.

The 2-D convolution defined in equ (7) can be viewed as a 1-D convolution as follows:



(a) 2-D convolution

$x = x_{11},x_{12},x_{13},x_{14},x_{15},x_{21},x_{23},x_{24},x_{25},x_{31},\ldots,x_{54},x_{55}$
$w = w_{11},w_{12},w_{13},0,0,w_{21},w_{22},w_{23},0,0,w_{31},w_{32},w_{33}$

(b) 1-D convolution

Figure (3) a) 2-D convolution, b) 1-D convolution, n = 5, m = 3

Following the same procedure, each row of the kernel can be represented as:

$$w_{i!} = w_{i1}, w_{i2}, \ldots, w_{im}, \qquad i = 1..m$$

and the kernel as:

$$w_{i!}, (n-m)![0], w_{2!}, (n-m)![0], \ldots, w_{m!}$$

where (n-m)![0] means a repetition of n-m zeros inserted to make each row of the kernel to be of the same length as the length of each row of the image matrix. Thus, the total of the 2-D kernel when transformed into a vector is $n(m-1)+m$.

At this point, we can relax the constraint stipulating that the input sequence is formed by the rows of the input array entered one after another into the systolic array without any delay between its consecutive rows.

Now, following Kung's method but rather than viewing the kernel as sliding over the image, we consider the kernel as stored in reverse order in the systolic array of $n(m-1)+m$ components one at every component from the right to the left, that is as:

$w_{33}, w_{32}, \ldots$

We also consider that the image is sliding from left to right at every pulse of the network. The y's are sliding at the same speed as the x's of the image but in the opposite direction, from right to left, and the $n(m-1)+m$ multiplications are performed simultaneously and stored in y.

Also, each processor has an input register to receive the x's, a register to hold the kernel value, a multiplier and an adder to receive the accumulated y's that result from its right neighbour and to add it to the result of its multiplication, then passes it to its left neighbour, see Figure (4). When the result accumulates its final value y_s, it will be outputted to host-1 at the left where it

$$y_s = \sum_{i,j=1}^{m} W_{i,j}\, X_{i+\lfloor s/(n-m+1) \rfloor,\, j+s \bmod (n-m+1)}$$

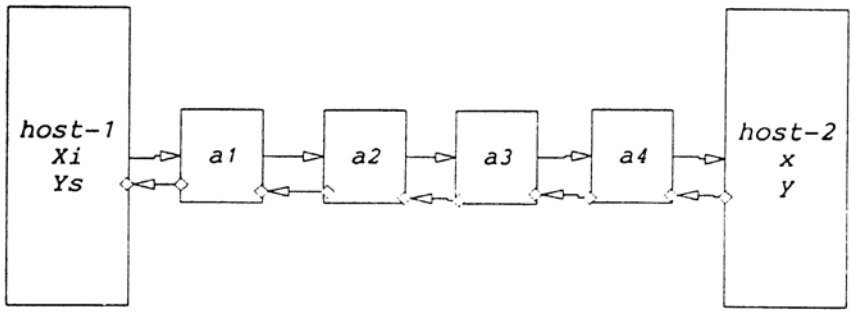


Figure (4). Systolic Array for Convolution

5.2 A New Parallel Algorithm for Low-Pass Digital Image Filtering:

5.2.1 Problems In the Old Algorithm:

can be easily seen that the old algorithm has a major problem when it reaches to the addition operations. The problem arises because of the addition operations are done at every component one after another, so no parallelism is performed in these operations. Following the same concept of computing the 2-D convolution by transforming it into a 1-D convolution and keeping the kernel stationary as before, it turns out that if the y's slide in either direction, then there is no gain in speed. This is due to the fact that after each of the $n(m-1)+m$ x's arrives at the corresponding component and all the $n(m-1)+m$ multiplications are carried out simultaneously, the addition operations will be carried out in a pipelining manner one after another. In other words, the addition operations will be performed sequentially.

5.2.2 Parallelism In The New Algorithm By Folding:

The new algorithm is based on the same concept as the old one except that rather letting the y's slide only from the opposite direction with respect to the x's we let the right half of the x's return from the most right component to the left after being multiplied. This is because of the left half of the PE's are sitting idle. In other words, to speed up the addition operations and hence the whole process and to make use of those PE's which are sitting idle, we divide the $n(m-1) + m$ components into two groups and let the most right component sends its result of multiplication to its left neighbor and the most left component sends its result to its right neighbor. During this operation, the additions are accumulated and then moved in both directions (we call this operation folding). This makes the additions to be performed in parallel in almost half the time required by the old design. When the two added final results of the two halves reach the extra middle component, one of them is delayed until the other is entered then the first one will be entered. This will require an additional time unit. After that they will be added up and the result of the addition is produced to the host after $\frac{1}{2}[n(m-1) + m] + 1$ time units which is almost half the time taken by the other method at the expense of an extra component, the middle component. See figure (5).

In figure (5), the x's slide left to right. the y's slide in two directions, $y_1 \ldots y_6$ slide from right to left and $y_7 \ldots y_{12}$ and y_{13} slide from left to right. As seen in figure (5) y_1 gets the result of multiplying x_{11} and w_{11} at the right most component and goes to the second component from the right in which the value of x_{12} is added to it. The same is done at the left side of the network where

```
w33 w32 w31 0 0 w23 w22 w21 0 0 w13 w12 w11

x11
x12 x11
x13 x12 x11
.   .   .   x11
.   .   .       .
.   .   .         .

x33 x32 x31 x25 x24 ....x15 x14 x13 x12 x11
```

Figure (5.a) The X's are shifted into the network

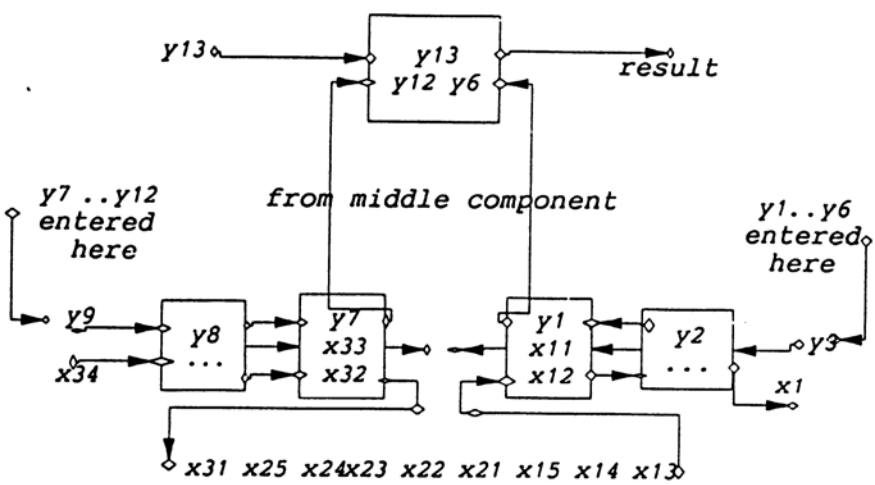


Figure (5.b) Addition by folding

Figure (5). New Algorithm with Systolic Array

y_7 gets the result of multiplying x_{33} and w_{33} at the left most component and goes to the second component from the left. When both of the accumulated results of y_6 and y_{12} come out of the two components surrounding the middle component they enter the extra (the one above the network in figure 5b) component to be

added and the result is then outputted. We would like to mention here that the x's are shifted off the network at the right and that the x_{11} in the second array from the right does not exist in reality but is shown for clarity. This value has already been multiplied and stored in y_1.

6. Performance And Comparison:

Consider the matrix of size nxn and the kernel of size mxm as above. Now, assume any operation takes one unit of time. Then in the old and the new algorithm, the elements of the matrix that are needed to be convoluted for the first time need nm-(n-m) time units to reach their corresponding components. Also one time unit for the multiplication is needed for the old and the new algorithms. Now, the addition and shifting will take another n(m-1) + m units in the old algorithm. Thus, the total number of time units needed in the old design is 2[n(m-1) + m] + 1. But in our new algorithm, the addition and the shifting need only ½[n(m-1) + m] + 1 units of time. This is because the two groups to be added are added at the same time in parallel, one group is added by a one half of the systolic array and the other group is added by the other half of the array. The extra time unit corresponds to the addition operation at the extra component of the systolic array. Thus, the total number of units needed in our new algorithm is only $\frac{3}{2}$ [n(m-1) + m] + 2. So, the total time for the whole process in the new algorithm is reduced to almost $\frac{3}{4}$ the time taken by the old algorithm.

7. Conclusion:

A new parallel algorithm using systolic array for a low-pass digital image filtering is presented. This new algorithm is faster and takes $\frac{3}{4}$ the time of the old algorithm at the expense of an extra component of the systolic array, assuming one unit of time for any operation. This can be implemented and tested by an Occam program that runs on a Transputer Development System (TDS). This work may be applied for the Laplacian and other high-pass filters.

References

[1] S. Yalamanchili, K. V. Palem, L. S. Davis, A. J. Welch, and J. K. Aggrawal, "Image processing architecture: A taxonomy and survey," in Progress in Pattern Recognition, vol. II. Amsterdam: North-Holland, 1985, pp. 1-37.

[2] M. J. B. Duff, "Review of the CLIp image processing system," in proc. Nat. comput. Conf. 19778, pp. 1055-1060.

[3] K.E. Batcher, "Design of a massively parallel processor, "IEEE Trans. on comput., vol. c-29, pp. 836-840, 1980.

[4] K. Luetjen, P. Gummar, and H. Ischen, "A fixible multi-processor system for image processing," in proc. 5th Int. Conf. Pattern Recognition, Miami Beach, Fl, Dec. 1980, pp. 326-328.

[5] P. Narendra, "VLSI architechure for real time processing," in Proc. Compcon, Spring 1981, pp. 303-306.

[6] S. R. Sternberg, "Architecture for neighborhood processing," in Proc. Pattern Recognition Image Processing Conf., Dallas, TX, Aug. 1981, pp. 374-380.

[7] John Bombardieni, "Systolic pipeline architecture for system convolution," IEEE Trans. on Signal Processing, vol. 40, no. 5, pp. 1253-1261, May 1992.

[8] Long-Wen Chang and Jin-Her Lin, "A bit level systolic array for median filter, IEEE Trans. on Signal Processing, vol. 40, no. 8, pp. 2079-2083, Aug. 1992.

[8] H. T. Kung, "Why systolic architectures," IEEE Computer

Mag., vol. 15, pp. 37-46, Jan. 1982.

[10] G. J. Li and W. Wah, "The design of optimal systolic arrays," IEEE Trans. on comp., vol. c-34, pp. 66-77, Jan. 1985.

[11] H. T. Kung and C. E. Leiserson, "Systolic array for VLSI, "in Proc. SIAm Sparse Matrix Proc., 1979, pp. 256-282.

[12] Colin D. Walter, "Systolic modular multiplication," IEEE Trans. on Comp., vol. 42, no.3, pp. 376-378, Mar. 1993.

[13] V. Boriakoff, "FFT computation with systolic arrays- A new architecture," IEEE Trans. on Circuits and Systems, II: Analog and Digital Signal Processing, vol. 41, n. 4, pp. 278-184, Apr. 1994.

[14] S. A. Amin, "Systolic design for lowpass digital image filtering on a transputer network using TDS," The SERC/DTI initiative in the engineering application of transputers, Mailshot, U.K. 1988.

[15] W. Niblack, An introduction to Digital Image Processing, U.K.: Printice-Hall Intr., 1986.

[16] A. K. Jain, Fundamentals of Digital Image Processing, Englewood Cliffs, NJ, Printice-Hall, In., 1989.

[17] R. C. Gonzalez and P. Wintz, Digital Image Processing, second edition, USA, Addison-Wesley, 1987.

[18] A. Rosenfield and A. C. Kak, Digital Picture Processing, vols, I and II, New York: Academic Press, 1976.

(Thesis, Columbia University, 1965).

[24] Ijiri, Y. And Simon, H.A., Effects of Mergers And Acquisions on Business Firm Concentration, Journal of Political Economy, 79 (March/April 1971), PP 314-322.

[25] Dewing, Arthur S., A Statistical Test of The Success of Consolidations, Quarterly Journal of Economics, Nov., 1921, PP. 84-101.

[26] Stigler, George J., The organization of Industry, Homewood, R.D. Irwin, 1968.

أبحاث جاهزة للنشر

١ - التكامل العمودي والأداء الاقتصادي في الصناعات الأردنية، د. وليد حميدات، رئيس قسم الاقتصاد، جامعة اليرموك: ٢٥ صفحة.

٢ - الآثار القانونية للعفو الخاص، عادل عبد إبراهيم، كلية الحقوق، جامعة مؤتة: ٣٢ صفحة.

٣ - وقف تنفذ العقوبة في القانون الأردني، د. محمد الجبور، كلية الحقوق، جامعة عمان: ٧٥ صحفة.

٤ - التشريع بين استحسان النقل واستحسان العقل، د. موسى أبو الريش، كلية الحقوق، جامعة عمان: ٤٥ صفحة.

٥ - العلاقة الدلالية بين إبليس والملائكة في ضوء قواعد باب الاستثناء، د.عودة أبو عودة، كلية الآداب والعلوم، جامعة عمان: ٣٠ صفحة.

٦ - ازدواج الجنسية في القانون الاردني والمقارن، د. غالب الداوودي، كلية الحقوق، جامعة اليرموك: ٣٠ صفحة.

٧ - Language Loyalty amongy the Yemenites of Lachawana / New York، د. بدر الدويك، قسم اللغة الانجليزية / جامعة عمان: ١٦ صفحة.

٨ - A New Parallel Algorithm for Lowpass Digital Image Filtering Using Systolic Array، د. عبد الرؤوف الحلاف، كلية العلوم، الجامعة الأردنية، د. سعد أمين / قسم الحاسوب / جامعة لوفبرة للتكنولوجيا: ٢٠ صفحة.

Interational Editions, Japan, 1984.

[13] Coase, R., The Nature of the Firm, Economica, New Series, Vol.4, 1937, 386-405.

[14] Bork, Robert, Vertical integration and the Sherman Act: The Legal History of Misconception 22 U. chi. L. Rev., 1954, 194-201.

[15] Machlup. Fritz, And Taber, Martha, Bilateral Monopoly, Successive Monopoly, and Vertical Integration, Economica, May, 1960, 101-119.

[16] Laffer, Arthur B., Vertical Integration By Corporations, 1929-1965, Review of Economics And Statistics, 51, Feb. 1969, 91-93.

[17] Gort, Michael, Testimony In The U.S Senate Antitrust Subcommittee Hearings, Economic Concentration, Part 2, 1965, 673-676.

[18] Adelman, M., Concepts and Statistical Measurement of Vertical Integration , In national Bureau of Economic Research, Businiess Concentration and Price Policy Princeton University Press, 1955, P.P 308-311).

[19] Thorp, W., The Integration of Industrial Operation (Wahington, D.C., 1924).

[20] Hart, P.E., Moment Distributions In Economics: An Exposition, Journal Of The Royal Statistical Society, Series A, 138 (Part 3), 1975, 73-85.

[21] Stigler, George J., Monopoly And Oligopoly By Merger, American Economic Review, XL, 2 (May 1950), 23-24.

[22] Reid, Samuel Richardson, Mergers And The Economy, Mc Graw - Hill, 1968.

[23] Kelly, Eamon, The profitability of Growth Through Mergers

References

[1] Shepherd, William G., The Economics of Industrial Organization, Prentice - Hall International, Inc., New Jersey, 1990.

[2] Barthwal, R.R., Industrial Economics, Wiley Eastern Limited, New Delhi, India, 1984.

[3] Clarkson, Kenneth W., And R.L. Miller, Industrial Organization: Theory, Evidence, And Public Ploicy, International Edition McGraw-Hill Book Co., Singapore, 1982.

[4] Sawyer, Malcolm C., The Economics of industries and firms, Croom Helm Ltd., Australia, 1985.

[5] Landsburg, Steven E., Theory And Applications, Harcourt Brace Jovanovich, U.S.A, 1992.

[٦] درويش، سليم، الاقتصاد الصناعي: تشكيلة - فعاليته وموقع المملكة العربية السعودية من تقنياته، تهامة، جدة، المملكة العربية السعودية، ١٩٨٥.

[7] Pickering, J.F., Industrial Structure and Market Conduct, Martin Robertson And Co. Ltd, London, 1974.

[8] Koch, James V., Industrial Organization And Prices, Prentice - Hall, Inc., Englewood Cliffs, New Jersey, 1974.

[9] Scherer, F.M., Industrial Market Structure and Economic Performance, Houghton Mifflin Company, Boston, 1980.

[١٠] نصر، محمد، التكامل العمودي في الصناعة الأردنية، مجلة دراسات (العلوم الإنسانية)، المجلد العشرون (أ)، العدد الثالث، ١٩٩٣.

[11] Gort, Michael, Diversification and Integration In American Industry, Princeton University Press, Princeton, 1962.

[12] Lindsay, Cotton M., Applied Price Theory, Holt-Saunders