# Generalized High-Level Synthesis of Wavelet-Based Digital Systems via Nonlinear I/O Data Space Transformations

Dongming Peng and Mi Lu

Electrical Engineering Department, Texas A&M University, College Station, TX77843, USA

### 1 Introduction

In this paper, we systematically present the high-level architectural synthesis for general wavelet-based algorithms via a model of I/O data space and the nonlinear transformations of the I/O data space. The parallel architectures synthesized in this paper are based on the computation model of distributed memory and distributed control. Several architectural designs have been proposed for the Discrete Wavelet Transform (DWT) [4]-[11]. None of these architectural designs for computing the DWT follows a systematic data dependence and localization analysis of general wavelet-based algorithms, and thus they only serve as particular designs and cannot be extended to other complicated wavelet-based algorithms such as MultiWavelet Transform (MWT)[1,13,14], Wavelet Packet Transform (WPT)[2,15] or Spacial-Frequential Quantization (SFQ) [12]. Using the WPT as a representative example of complex wavelet-based algorithms, this paper fully describes the theory and methodology used in synthesizing parallel architectures for general wavelet-based algorithms.

## 2 I/O Data Space Modeling of Wavelet-Based Algorithms

The basic equation for any discrete wavelet-based algorithms is generally represented by

$$\begin{split} X_{j+1}[t] &= \sum_{k \in L} C[k] X_j[Mt-k] \quad (Eq.1) \\ \text{where } C[k] \text{ are taps of a wavelet filter, } X_j \text{ and } X_{j+1} \text{ are the sequence of input} \\ \text{data and output data respectively at the } (j+1)^{th} \text{ level transform, } L \text{ is a set} \\ \text{that corresponds to the size of the wavelet filter, and } M \text{ is a constant scalar in} \\ \text{the algorithm. Generally, the algorithm is termed as } M\text{-}ary wavelet transform \\ \text{for } M \geq 2. \\ \text{There are } M \text{ wavelet filters for M-ary wavelet transform. If } X_j \text{ and } \\ X_{j+1} \text{ are scalar data and } C \text{ is scalar-valued taps of the wavelet filter, the algorithm is a classical scalar wavelet transform; if } X_j \text{ and } X_{j+1} \text{ are vector-valued } \\ \text{data and } C \text{ is matrix-valued taps of the multiwavelet filter, it is an MWT. If } t \\ \text{ and } k \text{ are scalars, the algorithm is a 1-D transform; if } t \text{ and } k \text{ are n-D vectors, it } \\ \text{is an n-D transform. Wavelet-based algorithms are multiresolution algorithms, } \\ \text{i.e., the output data at a level of transform can be further transformed at the next level.} \end{split}$$

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2073, pp. 884–893, 2001. © Springer-Verlag Berlin Heidelberg 2001

The following concepts are presented for the analysis in Section 3. Parameter index axis: The *parameter index axis* of a signal processing algorithm is the index axis for those data to be broadcasted in the algorithm, i.e., the data used in most computations but not generated by computations. As parameters of the computations, the number of them is fixed. Data index: The data index is the index for the intermediate data, input data, or output data that are generated and/or used by computations. In signal processing algorithms, the input size is variable. I/O data space: In an I/O data space the indexed data are only possibly the input data, output data or intermediate data, and the parameters of the algorithm are ignored. The intermediate data are viewed as partial inputs or partial outputs for the intermediate computations. Dependence graph: In this paper we present the I/O data space based dependence graph, where each node in the dependence graph corresponds to a data item, and each edge corresponds to a calculation or a dependence relation between the data used in the calculation and that generated in the calculation. Wavelet-adjacent field: In an I/O data space, a *wavelet-adjacent field* is a small domain made up of a group of source data items used by a calculation in Eq. (1). Its size is dependent on the wavelet filter. Super wavelet-dependence vector: A super waveletdependence vector  $d_{Wb}$  starts from a wavelet-adjacent field W and ends at the resulted data b. Since the source of the "dependence vector" itself is a domain instead of a single datum, we term such a dependence vector (corresponding to the calculation in Eq. (1) a super wavelet-dependence vector. In later analysis the super wavelet-dependence vectors are generally called dependence vectors and treated similarly as traditional dependence vectors. The length of a super wavelet-dependence vector  $|d_{Wb}|$  is defined as the Euclidean distance between a and  $b_c$ , where a and  $b_c$  are arithmetic centers of W and b respectively. **Regular dependence graphs:** in such dependence graphs the length of each dependence vector **d** is a constant value independent of either the input size or the data positions. Pseudo regular dependence graphs: in such dependence graphs the dependence vectors can be partitioned into a certain number of groups and in each group the length of dependence vectors is a constant value independent of either the input size or the data positions.

As examples, a wavelet-adjacent field, a super wavelet-dependence vector and the dependence graph for the algorithm of separable 2-D MWT[13][14] in I/O data space are shown in Figure 1 based on these concepts.

## 3 Nonlinear I/O Data Space Transformations for Regularizing Dependence Graphs

Theorem 3.1: The dependence graphs of wavelet-packet based algorithms (arbitrary expansion of wavelet trees) modeled in I/O data space can always be merged and regularized to a pseudo regular dependence graphs via appropriate nonlinear I/O data space transformations.



Fig. 1. The wavelet-adjacent  $\,$  eld, super dependence vector and the dependence graph for 2-D MWT  $\,$ 

#### Proof:

(1) For algorithms of 1-D transforms: There are M wavelet filters  $(f_1, f_2, \dots, f_M)$  at each level of 1-D M-ary wavelet transform, and each level of transform can decompose a certain subband into M components in wavelet-packet based algorithms. One of the filters  $(f_1)$  is for generating coarse component, others for detailed components. Assume M functions  $F_i(x) = Mx + i - 1$  for  $i = 1, 2, \dots, M$ .

Suppose that a subband  $\prod$  is calculated in l levels of wavelet-packet based transform consecutively with wavelet filters  $p_1, p_2, \dots, p_l$ , where  $p_u = f_i$  for  $u = 1, 2, \dots, l$  and i is any integer  $\in [1, M]$ . Considering that there are many subbands generated together by l levels of wavelet-packet based transform, and their corresponding dependence graphs should be merged as well as regularized to get a whole dependence graph for the algorithm, the nonlinear I/O data space transformation  $\Gamma_1$  is presented as follows. Without loosing generality, for the dependence graph corresponding to subband  $\prod$ ,  $\Gamma_1$  is:  $j \mapsto j; t \mapsto t$  if  $j = 0; t \mapsto P_1(P_2(\dots(P_j(t))\dots))$  otherwise, where  $P_u = F_i$  if  $p_u = f_i$  for  $u = 1, 2, \dots, j$ , and  $j \leq l, i \in [1, M]$ . Note that here j corresponds to the level of transform and can be only integers.

Consider another subband  $\prod_1$  different from  $\prod$  generated in the algorithm. Suppose that  $\prod_1$  is calculated in l levels consecutively with wavelet filters  $p'_1, p'_2, \dots, p'_l$ , where  $p'_u = f_i$  for  $u = 1, 2, \dots, l$  and i is any integer  $\in [1, M]$ . Since there exits at least one  $p'_u \neq p_u$  where  $u \in [1, l]$ ,  $\Gamma_1$  maps the data for  $\prod$  and  $\prod_1$  to different positions in the I/O data space. In other words,  $\Gamma_1$  can combine all dependence graphs of the subbands into a single I/O data space without conflicts.

Consider a dependence vector in the I/O data space corresponding to a calculation of Eq.(1) at data position  $t_0$  at the  $(u + 1)^{th}$  level of transform,

 $X_{u+1}[t_0] = \sum_{k \in L} p_{u+1}[k] X_u[Mt_0 - k],$ 

where  $p_{u+1}$  represents the wavelet filter used at this level. After the mapping of  $\Gamma_1$ , the calculation changes to

 $X_{u+1}[P_1(P_2(\cdots(P_u(P_{u+1}(t_0)))\cdots))] = \sum_{k\in L} p_{u+1}[k]X_u[P_1(P_2(\cdots(P_u(Mt_0 - k))\cdots))],$ 

where  $P_v = F_i$  if  $p_v = f_i$  for  $v = 1, 2, \dots, u+1$ , and  $i \in [1, M]$ . The dependence vector, starting from the wavelet-adjacent field which corresponds to L and is centered at data  $X_u[P_1(P_2(\cdots(P_u(Mt_0))\cdots))])$ , is targeted to data  $X_{u+1}[P_1(P_2(\cdots(P_u(P_{u+1}(t_0)))\cdots))]$ . The length of the dependence vector can be resolved by the difference between their coordinates along index j and t. The difference along index j is |u + 1 - u| = 1. The difference along t is  $|P_1(P_2(\cdots(P_u(P_{u+1}(t_0)))\cdots)) - P_1(P_2(\cdots(P_u(Mt_0))\cdots))| =$  $|P_1(P_2(\cdots(P_u(Mt_0+w))\cdots)) - P_1(P_2(\cdots(P_u(Mt_0))\cdots))| = M^u w$ , where w is an integer  $\in [1, M-1]$ . Here the length of the dependence vector is independent of  $t_0$ 's value, and the number of transform levels and the number of wavelet filters (M) remain constant in the algorithm. In other words, the lengths of the dependence vectors in the I/O data space after the mapping of  $\Gamma_1$  are bounded and independent of the data positions and input size. Moreover, the dependence vectors can be partitioned into a finite number of groups (according to the possible values of w and u), and the lengths of the dependence vectors in each group are the same. That is, the dependence graphs for 1-D wavelet-packet based algorithm are combined and regularized to be a pseudo regular dependence paragraph via the nonlinear I/O data space transformation  $\Gamma_1$ .

(2) For nonseparable n-D transforms: There are  $Q = M^n$  different wavelet filters  $(f_1, f_2, \dots, f_Q)$  at each level of n-D M-ary wavelet transform. Assume Q functions  $F_i(x) = Mx + q$ , where x and q are n-D vectors. The components of q are  $q_1, q_2, \dots, q_n$ , and  $q_v$  is an integer  $\in [0, M - 1]$  for  $v = 1, 2, \dots, n$ , and  $i = \sum_{u=1}^n M^u q_u$ . So  $i \in [1, Q]$ .

Suppose that a subband  $\prod$  is calculated in l levels of n-D packet-packet based transform consecutively with wavelet filters  $p_1, p_2, \dots, p_l$ , where  $p_u = f_i$  for  $u = 1, 2, \dots, l$  and  $i \in [1, Q]$ . Without loosing generality, for the dependence graph corresponding to subband  $\prod$ , a nonlinear I/O data space transformation  $\Gamma_2$  is presented as:  $j \mapsto j; t \mapsto t$  if  $j = 0; t \mapsto P_1(P_2(\dots(P_j(t))\dots))$  otherwise, where  $P_u = F_i$  if  $p_u = f_i$  for  $u = 1, 2, \dots, j$ , and  $j \leq l, i \in [1, Q]$ . Note that here j corresponds to the level of transform and can be only integers, and t represents n-D vectors.

For other subbands different from  $\prod$  generated in the algorithm, since there exits at least one filter used in the calculation of l levels of transform different from that of  $\prod$ ,  $\Gamma_2$  maps the data of them to different positions. In other words,  $\Gamma_2$  can combine all dependence graphs of the subbands into a single I/O data space without conflicts.

Similar to the case (1), a calculation corresponding to the dependence vector changes to

$$X_{u+1}[P_1(P_2(\cdots(P_u(P_{u+1}(t)))\cdots))] = \sum_{k \in L} p_{u+1}[k] X_u[P_1(P_2(\cdots(P_u(Mt-k))\cdots))],$$

where  $P_v = F_i$  if  $p_v = f_i$  for  $v = 1, 2, \dots, u+1$ , and  $i \in [1, Q]$ . The difference between the coordinates of the source and the target of the dependence vector along index j is |u + 1 - u| = 1. The difference along t is  $P_1(P_2(\dots(P_u(P_{u+1}(t)))\dots)) - P_1(P_2(\dots(P_u(Mt))\dots)) = P_1(P_2(\dots(P_u(Mt+1)))\dots)) = P_1(P_2(\dots(P_u(Mt+1))\dots)) = P_1(P_2(\dots(P_u(Mt+1))\dots) = P_1(P_2(\dots(P_u(Mt+1))\dots)) = P_1(P_2(\dots(P_u(P_u(Mt+1))\dots)) = P_1(P_2(\dots(P_u(P_u(Mt+1)))\dots)) = P_1$ 

#### 888 D. Peng and M. Lu

 $(w)(\cdots) - P_1(P_2(\cdots(P_u(Mt))\cdots)) = M^u w$ , where w is an n-D vector whose components are integers  $\in [1, M-1]$ . Thus the length of the dependence vector is independent of t's value. We have the similar conclusion that the lengths of the dependence vectors in the I/O data space after the mapping of  $\Gamma_2$  are bounded and independent of the data positions and input size, and the dependence vectors can be partitioned into a finite number of groups (according to the possible values of w and u), and the lengths of the dependence vectors in each group are the same.

(3) For separable n-D transforms: The n-D separable transforms are calculated separately and consecutively in every dimension. The index j is drawn in fractional numbers to represent the intermediate calculations in each level of transform. In the  $(s + 1)^{th}$  level (where s is an non-negative integer) of a separable n-D wavelet transforms, we have (n-1) intermediate I/O data planes  $j = s + 1/n, j = s + 2/n, \dots, j = s + (n - 1)/n$  between the planes j = s and j = s + 1. In the calculations for every dimension, there are M wavelet filters  $f_1, f_2, \dots, f_M$ , and a subband may be decomposed into M components on each dimension. So after each level of transform, a subband can be decomposed into  $M^n$  components. In addition, we assume M functions  $F_v(x) = Mx + v - 1$  for  $v = 1, 2, \dots, M$ .

Suppose that a certain subband  $\prod$  is calculated in l levels of n-D separable wavelet-packet based transform consecutively with wavelet filters  $p_{1,1}, p_{1,2}, \cdots, p_{1,n}, p_{2,1}, \cdots, p_{2,n}, \cdots, p_{l,n}$ , where  $p_{u,i} = f_v$  for  $u = 1, 2, \cdots, l$  and  $i = 1, 2, \cdots, n$ , and  $v \in [1, M]$ .  $p_{u,i}$  represents the wavelet filter used for the calculation of the  $u^{th}$ level transform on the  $i^{th}$  dimension in generating  $\prod$ . In order to regularize the dependence graphs, we present the nonlinear I/O data space transformation  $\Gamma_3$ as follows. Without loosing generality, for the dependence graph corresponding to subband  $\prod$ ,  $\Gamma_3$  is:  $j \mapsto j$ ,  $t_i \mapsto t_i$   $(i = 1, 2, \cdots, n)$  if j = 0;  $t_i \mapsto P_{1,i}(P_{2,i}(\cdots (P_{s+1,i}(t_i))\cdots))$  otherwise, with  $j \in [s+i/n, s+1+i/n)$ , s being an integer  $\in [0, l-1]$ ,  $P_{u,i} = F_v$  for  $p_{u,i} = f_v$   $(u = 1, 2, \cdots, s+1; i = 1, 2, \cdots, n;$ and  $v \in [1, M]$ ).

For other subbands different from  $\prod$  generated in the algorithm, since there exits at least one filter used in the calculation of l levels of transform different from that of  $\prod$ ,  $\Gamma_3$  maps the data of them to different positions.

The calculation of wavelet transform for the  $i^{th}$  dimension at the  $(s+1)^{th}$  level of transform for  $\prod$ ,

 $\sum_{k_i \in L_i} p_{s+1,i}[k_i] \sum_{k_{i+1} \in L_{i+1}} p_{s+1,i+1}[k_{i+1}] \cdots \sum_{k_n \in L_n} p_{s+1,n}[k_n] X_{s+(i-1)/n}[t_1, t_2, \cdots, Mt_i - k_i, Mt_{i+1} - k_{i+1}, \cdots, Mt_n - k_n],$ =  $\sum_{k_{i+1} \in L_{i+1}} p_{s+1,i+1}[k_{i+1}] \cdots \sum_{k_n \in L_n} p_{s+1,n}[k_n] X_{s+i/n}[t_1, t_2, \cdots, t_i, Mt_{i+1} - k_{i+1}, \cdots, Mt_n - k_n],$  $\cdots, Mt_n - k_n],$ 

changes to calculating

$$\begin{split} & \sum_{k_i \in L_i} p_{s+1,i}[k_i] \sum_{k_{i+1} \in L_{i+1}} p_{s+1,i+1}[k_{i+1}] \cdots \sum_{k_n \in L_n} p_{s+1,n}[k_n] X_{s+(i-1)/n}[P_{1,1}(P_{2,1}(\cdots (P_{s+1,1}(t_1)) \cdots )), P_{1,2}(P_{2,2}(\cdots (P_{s+1,2}(t_2)) \cdots )), \cdots, P_{1,i}(P_{2,i}(\cdots (P_{s,i}(Mt_i - k_i)) \cdots )), P_{1,i+1}(P_{2,i+1}(\cdots (P_{s,i+1}(Mt_{i+1} - k_{i+1})) \cdots )), \cdots, P_{1,n}(P_{2,n}(\cdots (P_{s,n}(Mt_n - k_n)) \cdots ))], \\ &= \sum_{k_{i+1} \in L_{i+1}} p_{s+1,i+1}[k_{i+1}] \cdots \sum_{k_n \in L_n} p_{s+1,n}[k_n] X_{s+i/n}[P_{1,1}(P_{2,1}(\cdots (P_{s+1,1}(t_1)) \cdots )), P_{1,2}(P_{2,2}(\cdots (P_{s+1,2}(t_2)) \cdots )), \cdots, P_{1,i}(P_{2,i}(\cdots (P_{s+1,i}(t_i)) \cdots )), P_{1,i+1}(P_{2,i+1}(\cdots (P_{s+1,i}(t_i)) \cdots ))), P_{1,i+1}(P_{2,i+1}(\cdots (P_{s+1,i}(t_i)) \cdots )), P_{1,i+1}(P_{2,i+1}(\cdots (P_{s+1,i}(t_i)) \cdots )), P_{1,i+1}(P_{2,i+1}(\cdots (P_{s+1,i}(t_i)) \cdots ))), P_{1,i+1}(P_{2,i+1}(\cdots (P_{s+1,i}(t_i)) \cdots ))))$$

 $(P_{s,i+1}(Mt_{i+1}-k_{i+1}))\cdots)$ ,  $\cdots$ ,  $P_{1,n}(P_{2,n}(\cdots(P_{s,n}(Mt_n-k_n))\cdots))$ ] after the mapping of  $\Gamma_3$ , where  $P_{u,i} = F_v$  if  $p_{u,i} = f_v$  for  $u = 1, 2, \cdots, s+1$ ;  $i = 1, 2, \cdots, n$ ; and  $v \in [1, M]$ . Thus after the transformation  $\Gamma_3$ , the dependence vector, starting from the

wavelet-adjacent field which corresponds to  $L_i$  and is centered at data  $X_{s+(i-1)/n}[P_{1,1}(P_{2,1}(\cdots(P_{s+1,1}(t_1))\cdots)), P_{1,2}(P_{2,2}(\cdots(P_{s+1,2}(t_2))\cdots)), \cdots, P_{1,i}(P_{2,i}(\cdots(P_{s,i}(Mt_i))\cdots)), P_{1,i+1}(P_{2,i+1}(\cdots(P_{s,i+1}(Mt_{i+1}-k_{i+1}))\cdots)), \cdots, P_{1,n}(P_{2,n}(\cdots(P_{s,n}(Mt_n-k_n))\cdots))],$ is targeted to data

$$\begin{split} X_{s+i/n}[P_{1,1}(P_{2,1}(\cdots(P_{s+1,1}(t_1))\cdots)),P_{1,2}(P_{2,2}(\cdots(P_{s+1,2}(t_2))\cdots)),\cdots,P_{1,i}\\ (P_{2,i}(\cdots(P_{s+1,i}(t_i))\cdots)),P_{1,i+1}(P_{2,i+1}(\cdots(P_{s,i+1}(Mt_{i+1}-k_{i+1}))\cdots)),\cdots,P_{1,n}\\ (P_{2,n}(\cdots(P_{s,n}(Mt_n\ -\ k_n))\cdots))]. \text{ The difference between the coordinates of the target and the source of the dependence vector along index $j$ is <math>|(s+i/n)-(s+(i-1)/n)|=1/n$. The difference along $t$ is } \end{split}$$

 $P_{1,i}(P_{2,i}(\cdots(P_{s,i}(P_{s+1,i}(t_i)))\cdots)) - P_{1,i}(P_{2,i}(\cdots(P_{s,i}(Mt_i))\cdots))$ 

=  $P_{1,i}(P_{2,i}(\cdots(P_{s,i}(Mt_i + w))\cdots)) - P_{1,i}(P_{2,i}(\cdots(P_{s,i}(Mt_i))\cdots)) = M^s w$ , where w is an integer  $\in [1, M - 1]$ . Thus the length of the dependence vector is independent of t's value. We have the similar conclusion that the lengths of the dependence vectors in the I/O data space after the mapping of  $\Gamma_3$  are bounded and independent of the data positions and the input size, and the dependence vectors can be partitioned into a finite number of groups (according to the possible values of w and s), and the lengths of the dependence vectors in each group are the same.  $\Box$ 

## 4 Design Example: Synthesis of 2-D WPT by Exploiting Inter-iteration Parallelism



Fig. 2. An instance of arbitrary wavelet tree expansion in the algorithm of 2-D WPT

The recursive separable 2-D WPT is illustrated in Figure 2 and the equations are as the followings

890 D. Peng and M. Lu

 $\begin{array}{l} C^{(j+1,4i)}[n_1,n_2] = \sum_{k1} \sum_{k2} h[k_1] h[k_2] \times C^{j,i}[2n_1-k_1,2n_2-k_2] \\ C^{(j+1,4i+1)}[n_1,n_2] = \sum_{k1} \sum_{k2} h[k_1] g[k_2] \times C^{j,i}[2n_1-k_1,2n_2-k_2] \\ C^{(j+1,4i+2)}[n_1,n_2] = \sum_{k1} \sum_{k2} g[k_1] h[k_2] \times C^{j,i}[2n_1-k_1,2n_2-k_2] \\ C^{(j+1,4i+3)}[n_1,n_2] = \sum_{k1} \sum_{k2} g[k_1] g[k_2] \times C^{j,i}[2n_1-k_1,2n_2-k_2] \\ \text{where } \mathbf{C}^{(j,i)}[n_1,n_2] \text{ means the datum at the position of } \mathbf{n}_1^{th} \text{ row, } \mathbf{n}_2^{th} \text{ column in the } i^{th} \text{ subband in transform level } j, h \text{ and } g \text{ are low- and high-pass wavelet filters. } \mathbf{C}^{(0,0)} \text{ is the input image, } L_0 \text{ is the wavelet filter length, } J \text{ is the highest transform level, and } \mathbf{N}^2 \text{ is the size of the input image.} \end{array}$ 

In Figure 2, the label for subband  $C^{(j,i)}$  generated in the  $j^{th}$   $(1 \le j \le J)$ level of WPT is given as a combination of H's and L's, which represent a binary number if we refer to H as "1" and L as "0". This binary number is equal to *i*. For example, subband  $C^{(2,14)}$  is labeled as HHHL in Figure 2, or "1110" (14 in decimal). Note that some components in a transform level may not be recursively decomposed in the next level WPT transform. According to  $\Gamma_3$  in Section 3, the nonlinear I/O index space transformation to merge all dependence graphs for generated subbands (as in Figure 2) is: (1) for the  $p_1p_2p_3.....p_{2m-1}p_{2m}$  subband in Figure 2 (result of the  $m^{th}$  level 2-D WPT),  $n_1 \mapsto P_2(P_4(...(P_{2m}(n_1))...)); n_2 \mapsto P_1(P_3(...(P_{2m-1}(n_2))...)); j = m; (2)$  for the  $p_1p_2p_3.....p_{2m}p_{2m+1}$  subband in Figure 2 (intermediate result in the  $(m+1)^{th}$ level 2-D WT),  $n_1 \mapsto P_2(P_4(...(P_{2m}(n_1))...)); n_2 \mapsto P_1(P_3(...(P_{2m+1}(n_2))...));$  $j = m + \frac{1}{2}$ , where we assume that function  $P_k(x)$  is Low(x) = 2x if  $p_k$  is "L"; or  $P_k(x)$  is High(x) = 2x + 1 if  $p_k$  is "H" for  $k = 0, 1, 2, \dots, 2m$ . Figure 3 shows the result of the nonlinear transformation in plane j=2 in the I/O data space.

In this section, we propose the parallel computing of 2-D WPT by exploiting the inter-iteration parallelism *based on the regularized and merged dependence* graphs via this nonlinear I/O data space transformation. The input, pixels of a 2-D image signal are assumed to be fed to multi-processors in parallel. The following concepts are adopted in the rest of this section.

Processor assignment: In this paper processor assignment is taken equivalently as I/O data space segmentation, where the I/O data space is segmented into subspaces, and the computations corresponding to the super dependence vectors in each segmented subspace are assigned to a processor.

Boundary dependence vectors vs. central dependence vectors: After segmenting the I/O data space, those dependence vectors lying in more than one subspaces of the I/O index space are called *boundary dependence vectors*, otherwise central dependence vectors.

Computation scheduling and permissible scheduling: the processor assignment is accompanied by a computation scheduling scheme, which specifies the order of the calculations in all the processors. A permissible schedule must satisfy two conditions: 1) the inherently sequential computations cannot be scheduled to the same time, i.e., the schedule cannot contradict the dependence graph; 2) no more than one computations can be performed in a processor at the same time.

*Boundary computation:* A processor's *boundary computations* are those performed in this processor and necessitating that the result of the computations be sent out to other processors.

#### Generalized High-Level Synthesis of Wavelet-Based Digital Systems 891

Row-column coordinates for data in different subbands after the nonlinear I/O data space transformation:

I	L: (2	2x,2y)										
I	HLL:	(4x,4y-	⊧1)									
I	HLH:	(4x,4y	+3)									
1	HHL:	(4x+2,	4y+1)									
L	литт.	(4x+2, (4x+1))	4y+3) 4y+1)									
ŀ	HILL.	(4x+1)	4y+1) 4y+3)									
ŀ	HHL:	(4x+3.	4v+1									
H	HHH:	(4x+3	4y+3)									
ł	IL: (2x	+1,2y)										
v	vhere x	and y	are no	n-nega	tive in	legers.						
Ŋ	(											
1												
2	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
1	LHLH	HHLH	LHHH	нннн	LHLH	HHLH	LHHH	нннн	LHLH	HHLH	LHHH	нннн
0	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
	LHLL	HHLL	LHHL	HHHL	LHLL	HHLL	LHHL	HHHL	LHLL	HHLL	LHHL	HHHL
	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
	LHLH	HHLH	LHHH	нннн	LHLH	HHLH	LHHH	нннн	LHLH	HHLH	LHHH	нннн
	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
	LHLL	HHLL	LHHL	HHHL	LHLL	HHLL	LHHL	HHHL	LHLL	HHLL	LHHL	HHHL
	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
	LHLH	HHLH	гннн	нннн	LHLH	HHLH	LHHH	нннн	LHLH	HHLH	LHHH	нннн
	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
	LHLL	HHLL	LHHL	HHHL	LHLL	HHLL	LHHL	HHHL	LHLL	HHLL	LHHL	HHHI
	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL	LL	HL
	0			2			6	7		8	0	10

Fig. 3. The data redistribution based on the I/O data space transformation for the 2-D WPT algorithm shown as in Figure 2

For briefing communication network and concentrating on the computation scheduling within processors based on the I/O data space transformation, we assume mesh-like processor (or PE) array to be used to implement the algorithm. For the purpose of minimizing data communication intensity, the total number of boundary dependence vectors is made as small as possible after the I/O data space segmentation. Thus, we segment the I/O data space in the direction of most dependence vectors, i.e., along the direction parallel with j axis. The shape of dependence graphs in each segmented subspace has the similarity as that in the whole I/O data space, but with different boundary dependence vectors generated in the segmentation. For the purpose of load balance among processors, all the subspaces are supposed to have the same size after the segmentation. Thus, we have partitioned the I/O data space into  $p^2$  subspaces as in Figure 4, where  $p^2$ is the number of parallel processors (or PE's).

To minimize the data communication intensity or the requirement on the network bandwidth, the boundary computations in a processor are scheduled as far away as possible in timing. A simpler explanation for this is that the result of a boundary computation is sent as soon as calculated and it is alright if the communication is completed before the result of the next boundary computation is generated and sent in the sense that the communication conflicts are avoided. In other words, to maximize the intervals between the boundary computations, each processor takes turns to execute one boundary computation and R non-boundary computations which involve boundary dependence vectors and central dependence vectors respectively, where R is the ratio of the number

#### 892 D. Peng and M. Lu



Fig. 4. The segmentation of I/O data space

of central dependence vectors to the number of boundary dependence vectors in the processor.

#### $\mathbf{5}$ Conclusions

This paper has demonstrated that data dependence analysis provides the basis for the synthesis of parallel architectural solutions for general wavelet-based algorithms and serves as a theoretical foundation for exploiting properties. Extracting the common features of computation locality and multirate signal processing within the wavelet-based algorithms, this paper contributes to data dependence and localization analysis based on a new concept - I/O data space analysis which leads to simplified structures of dependence graphs, and nonlinear I/O data space transformations for generalized high-level architectural synthesis of wavelet-based algorithms.

### References

- 1. M. Cotronei, L. B. Montefusco and L. Puccio, Multiwavelet analysis and signal processing, IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 45, Aug. 1998, pp. 970 -987.
- 2. F. G. Meyer, A. Z. Averbuch and J. O. Stromberg, Fast adaptive wavelet packet image compression, IEEE Transactions on Image Processing, Vol. 9, May 2000, pp. 792 -800.
- 3. J. Fridman and E. S. Manolakos, Discrete wavelet transform: data dependence analysis and synthesis of distributed memory and control array architectures, IEEE Transactions on Signal Processing, Vol. 45, May 1997, pp. 1291 -1308.
- 4. H. Sava, M. Fleury, A. C. Downton and A. F. Clark, Parallel pipeline implementation of wavelet transforms, IEE Proceedings on Vision, Image and Signal Processing, Vol. 144, Dec. 1997, pp. 355 -360.

- K. K. Parhi and T. Nishitani, VLSI architectures for discrete wavelet transforms, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 1, June 1993, pp. 191 -202.
- A. Grzeszczak, M. K. Mandal and S. Panchanathan, VLSI implementation of discrete wavelet transform, *IEEE Transactions on Very Large Scale Integration* (VLSI) Systems, Vol. 4, Dec. 1996, pp. 421 -433.
- Seung-Kwon Pack and Lee-Sup Kim, 2D DWT VLSI architecture for wavelet image processing, *Electronics Letters*, Vol. 34, March 1998, pp. 537 -538.
- G. Knowles, VLSI architecture for the discrete wavelet transform, *Electronics Letters*, Vol. 26, 19 July 1990, pp. 1184 -1185.
- M. Vishwanath, R. M. Owens and M. J. Irwin, VLSI architectures for the discrete wavelet transform, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol. 42, May 1995, pp. 305-316.
- T. C. Denk and K. K. Parhi, VLSI architectures for lattice structure based orthonormal discrete wavelet transforms, *IEEE Transactions on Circuits and Sys*tems II: Analog and Digital Signal Processing, Vol. 44, Feb. 1997, pp. 129-132.
- C. Chakrabarti and M. Vishwanath, Efficient realizations of the discrete and continuous wavelet transforms: from single chip implementations to mappings on SIMD array computers, *IEEE Transactions on Signal Processing*, Vol. 43, March 1995, pp. 759 -771.
- Zixiang Xiong, K. Ramchandran and M.T.Orchard, Wavelet packet image coding using space-frequency quantization, *IEEE Transactions on Image Processing*, Vol. 7, June 1998, pp. 892 -898.
- M. Cotronei, D. Lazzaro, L. B. Montefusco and L. Puccio, Image compression through embedded multiwavelet transform coding, *IEEE Transactions on Image Processing*, Vol. 9, Feb. 2000, pp. 184 -189.
- Gang Lin and Ze-Min Liu, The application of multiwavelet transform to image coding, *IEEE Transactions on Image Processing*, Vol. 9, Feb. 2000, pp. 270–273.
- F. Kurth and M. Clausen, Filter bank tree and M-band wavelet packet algorithms in audio signal processing, *IEEE Transactions on Signal Processing*, Vol. 47, Feb. 1999, pp. 549 -554.