Highly Efficient High-Speed/Low-Power Architectures for the 1-D Discrete Wavelet Transform

Francescomaria Marino, David Guevorkian, and Jaakko T. Astola

Abstract—In this paper, we propose two scalable architectures (say, Arc_J and Arc_L) that perform the discrete wavelet transform (DWT) of an N_0-sample sequence in only N_0/2 clock cycles. Therefore, they are at least twice as fast as the other known architectures. Also, they have an A_T^2 parameter that is approximately 1/2 that of already existing devices.

This result has been achieved by means of a carefully balanced pipelining, and it has two “factors.” First, Arc_J and Arc_L can be employed for performing two times faster processing than allowed by other architectures working at the same clock frequency (high-speed utilization). Second, they can be employed even using a two times lower clock frequency but reaching the same performance as other architectures. This second possibility allows for reducing the supply voltage and the power dissipation, respectively, by a factor of two and four with respect to other architectures (low-power utilization).

As a final result, we show that a parallel architecture implementing a L-tap filter-based DWT with J decomposition levels [say, Arc_O PT (J, L)] can be defined, aiming at having an excellent efficiency (say, eff[Arc_O PT (J, L)]) for any value of J and L. For instance, the average value of eff[Arc_O PT (J, L)] (computed in very wide set $\Sigma'$ of “points” (J, L)] is 99.1%. The minimum value of eff[Arc_O PT (J, L)] in $\Sigma'$ is 93.8%, and, except for five “points,” in all the others, eff[Arc_O PT (J, L)] is not lower than 96.9%.

I. INTRODUCTION

THE DISCRETE wavelet transform (DWT) [1]–[4] is a mathematical technique that decomposes a signal in the time domain by using dilated/contracted and translated versions of a single basis function, named the prototype wavelet. In the last decade, the DWT has often been found preferable to other traditional signal-processing techniques since it offers useful features such as inherent scalability, computational complexity of $O(N_0)$ (where $N_0$ are the samples of the processed sequence), low aliasing distortion for signal-processing applications, and adaptive time-frequency windows. Hence, the DWT has been studied and applied to a wide range of applications: numerical analysis [5], [6], biomedicine [7], different branches of image and video processing [1], [8], [9], signal-processing techniques [10], speech compression/decompression [11], etc.

In many of these applications, real-time performances are required in order to achieve attractive results. Therefore, the implementation of the DWT by means of dedicated VLSI application-specific integrated circuits has recently captivated the attention of a number of researchers, and many DWT architectures have already been proposed [12]–[25]. Some of these devices have been targeted to have a low hardware complexity, but they require at least 2N_0 clock cycles (ccs) to compute the DWT of a sequence $f^0$ having $N_0$ samples (e.g., the devices proposed in [12]–[14], the architecture A2 in [15], etc.). Nevertheless, also a large number of devices, having a period of approximately $N_0$ ccs, has been designed (e.g., the three architectures in [14] when they are provided with a doubled hardware, the architecture A1 in [15], the architectures in [16]–[18], the parallel filter in [19], etc.). Most of these architectures exploit the recursive pyramid algorithm (RPA) [26] or similar scheduling techniques in order both to reduce memory requirement and to employ only one or two filter units, independently from the number of decomposition levels to be computed. This is done by producing each output at the “earliest” instance that it can be produced [26].

The demand of low-power VLSI circuits in modern mobile/visual communication systems is growing all the time. On the other hand, the running progress of the VLSI technology has strongly reduced the cost of the hardware. Therefore, the possibility of reducing the period, even increasing the hardware, is becoming an important issue. In fact, low-period devices are important also for low-power utilization. For instance, a device $D$ having a period $T = N/2$ ccs could be employed not only for performing a two times faster processing than that allowed by $D'$ having a period $T' = N$ ccs but also (clocked at a frequency $f$) for reaching the same performance reached by $D'$, when this one is clocked at a frequency $f' = 2f$. This circumstance allows $D$ of reducing the supply voltage (linear in $f$) and the power dissipation (linear in $f^2$), respectively, by a factor of two and four with respect to $D'$ [27].

The above considerations have motivated the work of this paper, which proposes two scalable architectures having an A_T^2 parameter that is approximately 1/2 that of already existing devices and performing the DWT of an $N_0$-sample sequence in only $N_0/2$ ccs, since they allow the sampling of the input sequence at a three frequency two times higher than the working clock frequency. The proposed devices are a pipeline (namely, Arc_J) and a “hybrid pipeline-RPA” architecture (namely, Arc_L).

1The approach of merely using a team of $K$ devices that cooperate in order to speed up the computation by $K$ is not possible for most DWT applications, which require on-the-fly processing (i.e., to produce the output at the same rate that the input samples are received).
Fig. 1. One-dimensional DWT: dyadic decomposition in four levels. The symbol $\downarrow$ denotes decimation by two.

Even though the pipeline paradigm already has been exploited by existing DWT architectures (e.g., those in [12] and [23]–[25]), to the best of our knowledge, none of them reaches the performance claimed in this paper. Also, it is worth noting that the existing pipelined devices are underutilized. In fact, the downsampling occurring in each DWT decomposition level greatly helps in designing RPA-based architectures (e.g., those in [14]–[17], [19], [20], [22], etc.) but makes the pipelined devices heavily underutilized, since the stage implementing the decomposition level $k$ is usually clocked by a frequency $2^{k-1}$ times lower than the clock frequency used in the first level [24]. This underutilization comes from a low balancing of the pipelines when they implement the DWT and leads to a low efficiency. On the other hand, the architectures Arc$_J$ and Arc$_{\Sigma}$ are highly balanced.

In addition, because the designs of the proposed architectures depend only on the number of decomposition levels $J$ and on the length of the filter $L$, we characterize the efficiencies (otherwise said hardware utilization) of Arc$_J$ and Arc$_{\Sigma}$ within a 2-D space $\Sigma$ of coordinates $(J, L)$. Such characterization shows that the efficiency of Arc$_J$ decreases with $J$, while the efficiency of Arc$_{\Sigma}$ has an opposite behavior. This study suggests defining an architecture $\text{Arc}_{\text{OPT}}(J', L')$, which is highly efficient for any specific application identified by a point $(J', L') \in \Sigma$. $\text{Arc}_{\text{OPT}}(J', L')$ is simply the architecture (between Arc$_J$ and Arc$_{\Sigma}$) having the highest efficiency in $(J', L')$. The efficiency of $\text{Arc}_{\text{OPT}}$, evaluated for a very wide subset of points in $\Sigma$, has excellent results, 99.1% being the average value.

The outline of this paper is as follows. In the next section, the one-dimensional (1-D) DWT is shortly recalled. The strategy leading to the design of Arc$_J$ is illustrated in Section III. The computing blocks of Arc$_J$ implementing the decomposition levels 1 and 2 (namely, $B_2$ and $B_3$) are described, respectively, in Sections III-A and III-B. Designs of blocks implementing decomposition levels higher than the second one are described in Section III-C. The “hybrid pipeline-RPA” architecture Arc$_{\Sigma}$ is motivated and described in Section IV. Evaluations of the proposed architectures in terms of computing performances and efficiency, as well as the definition of Arc$_{\text{OPT}}$, are given in Section V. Conclusions are summarized in Section VI.

II. THE 1-D DISCRETE WAVELET TRANSFORM

The 1-D DWT [1]–[4] is a multilevel decomposition technique. In it, each decomposition level $j$ ($1 \leq j \leq J$; $N_0 = p2^d$, $N_0$, and $p$ being, respectively, the length of the input sequence $x^0$ and an integer) can be seen as the further decomposi-

\begin{align}
U_{n}^j &= \sum_{i=0}^{\left\lfloor L/2 \right\rfloor -1} a_{i} \cdot U_{n-2i}^{j-1}, \quad 0 \leq n < N_j \\
h_{n}^j &= \sum_{i=0}^{\left\lfloor L/2 \right\rfloor -1} c_{i} \cdot U_{n-2i}^{j-1}, \quad 0 \leq n < N_j
\end{align}

where $a_{i}$ and $b_{i}$ denote coefficients, respectively, of low-pass and high-pass $L$-tap filters (say, $a$ and $c$),$^2$ and we have assumed $l_{i} = 0$ for $n < 0$ or $n \geq N_j$.

A direct consequence of the decimation by two in (1a) and (1b) is the following Property P, which assumes particular importance in the body of this paper.

**Property P**

Let $U_{n}^{\text{EVEN}}$, $a_{n}^{\text{EVEN}}$, and $c_{n}^{\text{EVEN}}$ be the sequences consisting of the even-numbered samples, respectively, of $U_{n}^j$, $a_i$, and $c_i$ (i.e., $U_{n}^{\text{EVEN}} = U_{2n}^j$; $a_{n}^{\text{EVEN}} = a_{2n}$; and $c_{n}^{\text{EVEN}} = c_{2n}$; $0 \leq m < N_j/2; 0 \leq i < L/2$). Moreover, let $U_{n}^{\text{ODD}}$, $a_{n}^{\text{ODD}}$, and $c_{n}^{\text{ODD}}$ be the sequences consisting of the odd-numbered samples, respectively, of $U_{n}^j$, $a_i$, and $c_i$ (i.e., $U_{n}^{\text{ODD}} = U_{2n+1}^j$; $a_{n}^{\text{ODD}} = a_{2n+1}$; and $c_{n}^{\text{ODD}} = c_{2n+1}$; $0 \leq m < N_j/2; 0 \leq i < L - \left\lfloor L/2 \right\rfloor$). Therefore, $U_{n}^{j+1}$ and $h_{n}^{j+1}$ can be expressed as

\begin{align}
U_{n}^{j+1} &= \sum_{i=0}^{\left\lfloor L/2 \right\rfloor -1} a_{i}^{\text{EVEN}} \cdot U_{n-2i}^{j+1} \\
&\quad + \sum_{i=0}^{\left\lfloor L/2 \right\rfloor -1} a_{i}^{\text{ODD}} \cdot U_{n-2i}^{j+1} \quad 0 \leq n < N_{j+1}
\end{align}

\begin{align}
h_{n}^{j+1} &= \sum_{i=0}^{\left\lfloor L/2 \right\rfloor -1} c_{i}^{\text{EVEN}} \cdot U_{n-2i}^{j+1} \\
&\quad + \sum_{i=0}^{\left\lfloor L/2 \right\rfloor -1} c_{i}^{\text{ODD}} \cdot U_{n-2i}^{j+1} \quad 0 \leq n < N_{j+1}
\end{align}

$^2$The filters $a$ and $c$ may also have different number of taps. Nevertheless, in the literature on DWT architectures, they are considered to have an equal number of taps, since, in any case, the shorter filter can be suitably zero padded.
III. ARC $J$: A BALANCED PIPELINED APPROACH TO THE 1-D DWT

Because of its structure, 1-D DWT can be straightforwardly pipelined into $J$ blocks $B_j (1 \leq j \leq J)$, each block $B_j$ being devoted to compute the decomposition level $j$. Nevertheless, from (1a) and (1b), we observe that the complexity (say, $C_j$) of the decomposition level $j$ is linear in $N_j$ with a factor depending on $L$. Therefore, because of the decimation by two performed in each decomposition level

$$ C_j = 2C_{j+1}. \quad (2) $$

As a clear consequence of (2), in order to balance a pipelined DWT architecture, each block $B_j$ should employ a number $W_j$ of processing elements (PEs) (each PE will be basically constituted by one multiplier and one adder)

$$ W_j = 2W_{j+1}. \quad (3) $$

Therefore, our idea is to build a balanced architecture $Arc_J$ constituted by a pipeline $\{ B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_J \}$. Each block $B_j$ is designed carefully taking into account (3). A top-level scheme of $Arc_J$ is given in Fig. 2.

We assume $W_1 = 2L$ since, as we shall see in Section III, this choice leads to designs of a block $B_1$ having a period of only $N_0/2$ ccs that is 100% efficient and scalable with any value of $L$. Because of this choice, in order to balance Arc$_J$, from (3), we must have $W_2 = L$ and, in general

$$ W_k = \left\lceil \frac{L}{2^{k-2}} \right\rceil \quad (4) $$

where $\lceil \cdot \rceil$ denotes the rounding to the smallest integer not smaller than $\cdot$, and takes into account the discrete nature of the PEs.

Designs of blocks $B_1$ and $B_2$ employing, respectively, $2L$ and $L$ PEs, are introduced in Section III-A and -B, respectively. Blocks $B_k (3 \leq k \leq J)$ having $W_k$ PEs as defined in (4) are described in Section III-C.

A. First Level of Decomposition: $B_1$

In order to design a high-speed block $B_1$ performing the first level of DWT decomposition, we consider Property P introduced in Section II. For $j = 0$, Property P means that $\mathbf{f}[h^1]$ can be computed by the point-wise sum of two different and independent convolutions having, as filters, $a_{EVEN}^{EVEN}$ and $a_{ODD}^{ODD}$ [$c_{EVEN}^{EVEN}$ and $c_{ODD}^{ODD}$] and, as input, $f^{EVEN}$ and $f^{ODD}$, respectively.

Therefore, four independent filter units (say, $A^{EVEN}_{EVEN}$, $A^{ODD}_{EVEN}$, $C^{EVEN}_{EVEN}$, and $C^{ODD}_{EVEN}$), globally employing $2L$ PEs [say, $w_{a(0)}$, $w_{a(1)}$, $\ldots$, $w_{a(L-1)}$, $w_{c(0)}$, $w_{c(1)}$, $\ldots$, and $w_{c(L-1)}$], can be designed and arranged in order to process in parallel $f^{EVEN}$ and $f^{ODD}$ and therefore, perform the first level of DWT at the rate of 2 samples per cc, supposing that 1 cc is the time needed by one PE to compute one product and one sum.$^3$ Block $B_1$ is shown in Fig. 3 for $L = 6$. The data dependence graph (DDG) of $A^{EVEN}_{EVEN}$ and $A^{ODD}_{EVEN}$ computing $\mathbf{f}$ is provided in Table I. From this, the DDG of $C^{EVEN}_{EVEN}$ and $C^{ODD}_{EVEN}$ can be straightforwardly derived. In this DDG, and in the other DDGs that will be considered in the following, the columns related to the I/O data lines [e.g., $I_E$, $I_O$, $I_{O1}$, $I_{O1}(k-1)$, $O$, $O_{L(k)}$, and $O_{H(k)}$], as well as the columns related to the select signals (e.g., $S$, $S'$, $S''$, $S'''$), show the data present on that line in a specific cc [identified by the clock cycles counter $\tau(k)$, $1 \leq k \leq J$]. The products computed in each cc by the multipliers inside each PE are shown in the related columns. Arrows are used to denote how these products have to be added in order to achieve the final results.

Fig. 2. Arc$_J$: top-level architectural scheme.
Fig. 3. Block $B_1$: architectural scheme ($L = 6$). The gray latches have no functional reason but have been inserted in order to make the critical paths limited to one multiplier and one adder.

### Table I

<table>
<thead>
<tr>
<th>$I \cap \cap$</th>
<th>$I \cup \cup$</th>
<th>$I \cap \cup$</th>
<th>$I \cup \cap$</th>
<th>$I \cap \cap$</th>
<th>$I \cup \cup$</th>
<th>$I \cap \cup$</th>
<th>$I \cup \cap$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O_{\cap \cap}$</td>
<td>$O_{\cup \cup}$</td>
<td>$O_{\cap \cup}$</td>
<td>$O_{\cup \cap}$</td>
<td>$O_{\cap \cap}$</td>
<td>$O_{\cup \cup}$</td>
<td>$O_{\cap \cup}$</td>
<td>$O_{\cup \cap}$</td>
</tr>
</tbody>
</table>

#### B. Second Level of Decomposition: $B_2$

When more than one level of decomposition is needed, as in most of the applications, the sub-band $I$ produced by $B_1$ has to be further transformed. This requires a second block $B_2$, which has to be pipelined to the output line $O_{\cap \cap}$ coming from $B_1$. In this section, we describe how such a block $B_2$, having 100% efficiency for any value of $L$, can be designed.

As previously stated, $B_2$ has to be provided with $L$ PEs $w_i (0 \leq i < L)$ in order to constitute a balanced team with $B_1$. Therefore, a possible way of employing these PEs is to assign to $w_i$ the computations related both to $a_2$ and to $c_2$. This strategy is made possible by Property P introduced in the previous section.

In fact, as shown in the DDG of $B_1$ (Table I), the line $O_{\cap \cap}$ outputs the samples of $I$ in a sequential data stream at a rate of 1 sample per cc. Therefore, by splitting $O_{\cap \cap}$ into two different lines $I_E$ and $I_O$ carrying, respectively, $I^{EVEN}$ and $I^{ODD}$ into $B_2$, we have the possibility of duplicating on $I_E[O_I]$ the single sample $B_{2n}[I^{EVEN}][0 \leq n < N/2]$ before a new sample $B_{2n+1}[I^{ODD}]$ is sent by $O_{\cap \cap}$ onto $I_E[O_I]$, without increasing the period. By this way, $I_{2n}[I_{2n+1}]$ is available for 2 ccs and $u_{2m}[u_{2m+1}][0 \leq m < L/2]$ has the time for processing it both by $a_2n$ and by $c_{2m}$ [by $a_{2m+1}$ and by $c_{2m+1}$]. In order to do this, an Adapter such as that shown in Fig. 4 has to be inserted between $O_{\cap \cap}$ and the lines $I_E/I_O$. Such an adapter allows the following strategy:

\[
I_E(\tau(2)) = \begin{cases} 
O_{\cap \cap}(\tau(2)), & \text{if } S(\tau(2)) = 0 \\
I_E(\tau(2) - 1), & \text{if } S(\tau(2)) = 1 
\end{cases} \quad (5a)
\]

\[
I_O(\tau(2)) = \begin{cases} 
I_O(\tau(2) - 1), & \text{if } S(\tau(2)) = 0 \\
O_{\cap \cap}(\tau(2)), & \text{if } S(\tau(2)) = 1 
\end{cases} \quad (5b)
\]

where the clock cycles counter $\tau(2)$ is here adopted only in order to make transparent the propagation delay introduced in the pipe by $B_1$ [i.e., $\tau(2) = 0$ when $B_3$ is output by $B_2$] and $S(\tau(2))$ is a select signal, which is zero [1] for the even [odd] values of $\tau(2)$.

The functionality of $B_2$ becomes clear by an analysis of its DDG, which is given in Table II. Starting on $\tau(2) = 0$, the input line $I_E[O_I]$ provides the even-numbered [odd-numbered] PEs of $B_2$ with the even-numbered [odd-numbered] samples of $I^{EVEN}$, which are duplicated as shown in the $I_E[O_I]$ column of the DDG. The multipliers in the even-numbered PEs $u_{2m}$ are provided with two-cell circular shift registers (CSRs) pre-loaded in such a way that the filter coefficients $a_{2m}$ are used in the even-numbered ccs ($S = 0$) and the filter coefficients $c_{2m}$ are used in the odd-numbered ccs ($S = 1$). Conversely, CSRs connected in input to the multipliers of the odd-numbered PEs $u_{2m+1}$ operate in the opposite way, i.e., the coefficients of $a$ are used in the odd-numbered ccs, and the coefficients of $c$ are used in the even-numbered ccs. By this way, $I^2$ and $I^2$ can be pro-

\[\text{An alternative way of achieving the desired data flow onto } I_E \text{ and } I_O \text{ is to replace the two multiplexers in the Adapter with two latches triggered at a rate twice slower than the clock signal.}\]

\[\text{In practice, these circular shift registers work as multiplexers, but they are more compact. Their use, instead of the use of multiplexers, will be even more convenient in the other levels of the pipe, since, as we shall show in Section III-C, each multiplier in } B_k \text{ will need a } (2^{k-1})\text{-input multiplexer, suitably replaced by a } (2^k)\text{-cell CSR.}\]
Fig. 4. Block B₂: architectural scheme (L = 6). The select signal S, controlling multiplexers and demultiplexers, is not explicitly shown. The two-cell CSRs storing the filter coefficients work as two-input multiplexers.

TABLE II
BLOCK B₂: DDG (FOUR PERIODS)

<table>
<thead>
<tr>
<th>t</th>
<th>O₁(t)</th>
<th>O₂(t)</th>
<th>S₁(t)</th>
<th>S₂(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>f₀</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>f₁</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>f₂</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>f₃</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>f₄</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>f₅</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>f₆</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>f₇</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The select signal, controlling multiplexers and demultiplexers, is not explicitly shown. The two-cell CSRs storing the filter coefficients work as two-input multiplexers.

TABLE III
BLOCK B₃: DDG (THREE PERIODS). "–" DENOTES THAT ONLY PARTIAL RESULTS ARE AVAILABLE ON O

<table>
<thead>
<tr>
<th>t</th>
<th>O₁(t)</th>
<th>O₂(t)</th>
<th>S₁(t)</th>
<th>S₂(t)</th>
<th>S₃(t)</th>
<th>w₂</th>
<th>w₃</th>
<th>w₄</th>
<th>O₁(t+1)</th>
<th>O₂(t+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>f₀</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>f₀</td>
<td>f₀</td>
<td>0</td>
<td>f₀</td>
<td>f₀</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>f₁</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>f₁</td>
<td>f₀</td>
<td>0</td>
<td>f₁</td>
<td>f₀</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>f₂</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>f₂</td>
<td>f₀</td>
<td>0</td>
<td>f₂</td>
<td>f₀</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>f₃</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>f₃</td>
<td>f₀</td>
<td>0</td>
<td>f₃</td>
<td>f₀</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>f₄</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>f₄</td>
<td>f₀</td>
<td>0</td>
<td>f₄</td>
<td>f₀</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>f₅</td>
<td>f₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>f₅</td>
<td>f₀</td>
<td>0</td>
<td>f₅</td>
<td>f₀</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The select signal, controlling multiplexers and demultiplexers, is not explicitly shown. The two-cell CSRs storing the filter coefficients work as two-input multiplexers.
Fig. 5. Block \( B_3 \): architectural scheme \( (5 \leq L \leq 6) \). The Adapter can be replaced by a single latch triggered by a signal having a frequency twice as low as the frequency of the clock. The four-cell CSRs storing the filter coefficients work as four-input multiplexers.

The correct use of the filter coefficients by each multiplier is simply achieved by storing the coefficients in \( (2^{k-1}) \)-cell CSRs (one CSR for each PE) in the above-specified order. For \( k = 3 \), the operations performed in each period by \( u_{0} \) are simply those related to \( a_{2} \) and \( c_{2} \) (semiperiod \( S = 0 \)) and those related to \( a_{2i+1} \) and \( c_{2i+1} \) (semiperiod \( S = 1 \)). This scheduling takes into account Property P. The final results are produced at the same rate of the input, i.e., two samples per period, which means two samples every \( 2^{k-1} \) ccs (in this case, 4 ccs). The architecture implementing \( B_3 \) according to this data DDG is shown in Fig. 5.

The Adapter duplicates for one semiperiod, i.e., for \( 2^{k-2} \) ccs (for \( k = 3, 2 \) ccs) the input samples onto the line \( l \), as soon as they are produced by \( B_{k-1} \). The multiplexers in input to the adders select (by means of the signal \( l \)) the feedback or the systolic communication among the processors.

The proposed scheme assumes a substantial difference for \( k > 3 \). In fact, since \( u_{0} \) has to be shared by \( 2^{k-3} \) coefficients (which are at least two, for \( k > 3 \)) for each one of the filters \( a_{\text{EVEN}}, a_{\text{ODD}}, c_{\text{EVEN}} \) and \( c_{\text{ODD}} \), the same input data will be multiplied (in each PE) at least by two low-pass and at least by two high-pass filter coefficients. Therefore, the composition of the final result needs an external feedback between the first and the last PE of the systolic array. These concepts are clearer from an exam of the DDG of block \( B_3 \), which is provided in Table IV. The above-mentioned external feedback (denoted by gray arrows in the graphs) concerns \( (2^{k-2}-2) \) every \( 2^{k-2} \) data accumulated by the adder in \( u_{0} \) in the first semiperiod \( S = 0 \) (the remaining two data constitute a sample of \( \tilde{f} \) and a sample of \( h^{k} \) and do not need further processing). Specifically, these data are delayed by \( (2^{k-2}-2) \) ccs and sent in input to the adder of the last PE of the array (i.e., \( u_{W_k-1} \)). Therefore, in the semiperiod \( S = 1 \), the adder in \( u_{W_k-1} \) receives the data that were generated in \( u_{0} \) during the previous semiperiod. This external feedback is enabled by the combination \( \{S = 1, S' = 0\} \) and does not involve the last 2 ccs of the semiperiod \( S = 1 \), since in those ccs \( u_{W_k-1} \) is employing the filter coefficients having index \( (L_k - 1) = (2^k - 2)W_k - 1 \).

The architecture that realizes such a data graph is represented in Fig. 6. From the presented examples and from the above descriptions, designs of \( B_k \), with \( k > 4 \), can be easily derived.

Also, the extension to \( B_k \) having higher values of \( W_k \) is trivial. In the case of \( W_k = 1 \) (i.e., \( L = L_k = 2^k - 2 \)), only one PE is employed, and therefore, the external loop connecting the first PE with the last one becomes a loop connecting the only \( u_{0} \) with itself. Note that the Adapters at the input of the blocks shown in Figs. 5 and 6, as well as the tree of demultiplexers at the output of the blocks, can be avoided by a suitable sampling of the data lines.

### IV. Arc2: A Hybrid Pipeline-RPA Architecture for the 1-D DWT

As we shall show in the next section, Arc2 is 100% efficient for any value of \( L \) when \( J < 3 \). For \( J \geq 3 \), the efficiency of Arc2, say, \( \text{eff}[\text{Arc2}] \), is still 100%, but only if

\[
L = L_k(\equiv W_k \cdot 2^{k-2})
\]
for any $k \leq J$. This restriction is due to the rounding in (4) that leads to implementation in $B_k$ two filters having $L_k$ taps, where $L_k$ may also be greater than $L$. As we have seen in Section III, when (6) is not verified, $2(L_k - L)$ filter coefficients, among those preloaded in $B_k$, have to be zeroes and therefore, $B_k$ will be underutilized.

Another consequence of the rounding in (4) is that some $J$ and some $L$ may exist such that

$$\sum_{k=2}^{J} W_k = \sum_{k=2}^{J} \left\lfloor \frac{L}{2^{k-2}} \right\rfloor > 2L \tag{7}$$

which means that the blocks $B_2, B_3, \ldots, B_{J-1}$, and $B_J$ globally require more than $2L$ PEs.

On the other hand, from (2) in Section III, we observe that

$$C_2 > \sum_{k=2}^{J} C_k \tag{8a}$$

for any $1 \leq \tilde{j} < J$. In particular, for $\tilde{j} = 1$

$$C_1 > \sum_{k=2}^{J} C_k \tag{8b}$$

Therefore, it seems reasonable to us to design also a second architecture (say, Arc2), which is constituted by a pipeline \{ $B_1 \rightarrow B_2^2$ \} having only two blocks for any value of $J > 2$. In Arc2, the decomposition level 1 is produced by $B_1$ exactly as in Arc1, while all the levels $k(2 \leq k \leq J)$ are performed by a new block $B_2^2$. Because of (8b), a number $W_2^2$ of PEs not bigger than $W_1$ is needed by $B_2^2$ in order to not deteriorate the performance allowed by $B_1$. Therefore, we choose $W_2^2 = W_1 = 2L$.

As an effect of this choice, Arc2 will be more efficient than Arc1 in any application requiring $J$ decomposition levels and employing $L$-tap filter bases when $J$ and $L$ satisfy (7).

Fig. 7. Arc2: top-level architectural scheme.

The design of Arc2 does not require additional descriptive details, since we observe that the $N_1$ samples of $\hat{f}$ are fed into $B_2^2$ by the line $O_{L_{(1)}}$ in a sequential data stream and at the rate of 1 sample per cc. This means that $B_2^2$ can simply be an RPA-based architecture able to decompose in $J-1$ levels a sequence of $N_1$ samples in $N_1$ cc's, employing $2L$ PEs. Many of these devices have already appeared in the literature: for instance, the architectures described in [16]–[18], the parallel filter proposed in [19], the architecture “A1” proposed in [15], the three devices described in [14] when, as the same authors suggest, they are provided with a double number of processors, etc.

Therefore, for our purposes, a scheme of Arc2 can be simply proposed in the form of Fig. 7, where $B_1$ is the block described in Section III-A and $B_2^2$ is any RPA-based architecture employing $2L$ PEs and transforming the $N_1$ samples of $\hat{f}$ in $N_1$ cc's. It is worth noting that Arc2 (i.e., the above-described “coupled use” of $B_1$ with any already known RPA device $B_2^2$) allows two times faster processing than does the classical RPA-based device $B_2^2$. 
V. Performance Evaluations and Definition of \( \text{ARC}_{OPT}(J, L) \)

A. Period and \( AT^2 \) Parameter

In order to evaluate the global time required by the proposed architectures to perform the DWT of a sequence \( \mathbf{f} \) having \( N_0 \) samples, we observe that each block \( B_k \) has a period \( T_k = 2^k-1 \) ccs, since it produces a sample \( f_k \) every \( 2^k-1 \) ccs (immediately followed by \( h_k \)). As a consequence, each block needs \( T_k N_k / 2 = N_0 / 2 \) ccs to perform its task (balanced computation). Therefore, the period of \( \text{ARC}_J \) is

\[
T[\text{ARC}_J] = \frac{N_0}{2} + \Delta_J \quad [\text{ccs}]
\]

where \( \Delta_J \) takes into account the “startup” delay due to the propagation through the pipe (i.e., the number of ccs needed by \( B_j \) to output \( h_j \)). Nevertheless, it can be neglected since \( \Delta_J \approx J \), which, in practical applications, is very much smaller than \( N_0 \). Approximately \( N_0 / 2 \) ccs is also the value of \( T[\text{ARC}_J] \), when \( B_2 \) is implemented by devices as those referenced in Section IV.

Measures of \( T[\text{ARC}_J] \) and \( T[\text{ARC}_J] \) can be known only taking into account the particular adopted technology. However, to be independent of the technology, in a very first approximation (considering a latch inserted between each pair of adjacent blocks), we can assume the cc period as the latency of one multiplier plus the latency of one adder, which is the same assumption made by other authors. Therefore, as we have already claimed in the introduction, \( \text{ARC}_J \) and \( \text{ARC}_J \) are approximately two times faster than all the other “single chip” known architectures.

Our architectures allow approximately the same improvement (i.e., by a factor of two) in terms of the \( AT^2 \) parameter, as summarized in Table V. In such a table, the VLSI area has been characterized only by means of the number of multipliers and adders: such a measure is therefore only indicative. Nevertheless, it should be remarked that in DWT applications, the required precision grows with the levels. Therefore, while the only processing unit in classical RPA-based devices must achieve the precision required by the level \( J \) (i.e., the highest one), pipelines might be realized by blocks that in lower levels achieve lower precision. This possibility could provide further reduction to the VLSI area of \( \text{ARC}_J \) and \( \text{ARC}_J \). Moreover, \( \text{ARC}_J \) (and \( \text{ARC}_J \), in part) are semisystolic (i.e., they do not require complex routing, as many RPA devices do) and use a controller simpler than that needed by fully RPA architectures. In fact, the multiplexers and demultiplexers employed in any block \( B_k \) of \( \text{ARC}_J \) need a set of \( (k-1) \) select signals, which are the \( (k-1) \) least significant outputs of a counter modulo \( 2^{k-1} \). Therefore, this counter is the only control unit needed by \( \text{ARC}_J \), since it provides all the needed select signals.

B. Efficiency

Another important issue in parallel architectures is the effective utilization of the processors, which is characterized by the efficiency. We define the efficiency of a parallel architecture \( A \) having \( W[A] \) PEs as the ratio between the period \( T[B] \) of a nonparallel architecture \( B \) (i.e., \( W[B] = 1 \)) and \( W[A] \) times the period \( T[A] \) of the parallel architecture \( A \). From the efficiency \( \text{eff}[A] \), the speedup of an architecture \( A \) (say, \( \text{speedup}[A] \)) can be straightforwardly derived as \( \text{speedup}[A] = \text{eff}[A] W[A] \).

In order to simplify the following analysis, we define one product and one sum as one “basic computation” occurring in the 1-D DWT. Since we assumed 1 cc as the time needed by a single PE to perform one “basic computation,” the period \( T[B] \) measured in ccs is simply the number of basic computations required by the 1-D DWT. Because the production of each coefficient of a given subband requires \( L \) basic computations,\(^9\) we have

\[
T[B] = L \sum_{j=1}^{J} \frac{N_j}{2^{j-1}} \quad [\text{ccs}],
\]

On the other hand, the global number of PEs employed by \( \text{ARC}_J \) and \( \text{ARC}_J \) is\(^10\)

\[
W[\text{ARC}_J] = \sum_{j=1}^{J} \left[ \frac{L}{2^j} \right],
\]

\[
W[\text{ARC}_J] = 4 L \quad \text{for } J \geq 4.
\]

Therefore, considering \( T[\text{ARC}_J] = T[\text{ARC}_J] = N_0 / 2 \) ccs, \( \text{eff}[\text{ARC}_J] \) and \( \text{eff}[\text{ARC}_J] \) result

\[
\text{eff}[\text{ARC}_J] = \frac{\sum_{j=1}^{J} \left[ \frac{L}{2^{j-2}} \right]}{\sum_{j=1}^{J} \left[ \frac{L}{2^{j-2}} \right]} \quad \text{for } J \geq 4.
\]

\[
\text{eff}[\text{ARC}_J] = \sum_{j=1}^{J} \frac{1}{2^j} \quad \text{for } J \geq 4.
\]

\(^9\)Actually, any subband coefficient requires \( L \) products and \( L-1 \) sums, but we approximate these operations to \( L \) basic computations. On the other hand, this approximation compensates the approximation made in the opposite sense, when blocks \( B_k \) having \( W_k \) multipliers and \( W_k - 1 \) adders were considered having \( W_k \) PEs\(^9\).

\(^{10}\)\( \text{ARC}_J \) has to be considered only for \( J \geq 4 \). In fact, \( \text{ARC}_J \), for \( J < 4 \) and for any \( L \geq 2 \), requires fewer PEs than \( \text{ARC}_J \) does.
As was expected, the only parameters that affect the efficiencies are $J$ and $L$ (in particular, $\text{eff}[^{\text{Arc}_J}]$ depends only on $J$), since the designs of both the architectures are independent on $N_0$. Therefore, we can characterize any specific problem or application by the coordinates $(J, L)$ of a 2-D space $\Sigma$, and plot $\text{eff}[^{\text{Arc}_J}]$ and $\text{eff}[^{\text{Arc}_L}]$ for any point of $\Sigma$. Fig. 8(a) and (b) shows such a study for a wide subset $\Sigma' = \{(J, L), 1 \leq J \leq 8, 2 \leq L \leq 22\} \subset \Sigma$. $\text{eff}[^{\text{Arc}_J}]$ is 100% in all the points $(1, L)$ and $(2, L)$, which are denoted by “x” in Fig. 8(a). Nevertheless it is worthwhile to note the following.

1) For a given value of $L$, $\text{eff}[^{\text{Arc}_J}]$ decreases as $J$ increases.
2) $\text{eff}[^{\text{Arc}_L}]$ is independent on $L$ and increases with $J$.

In other words, $\text{Arc}_J$ and $\text{Arc}_L$ compensate themselves each other in terms of efficiency. This circumstance is visually evident in Fig. 8(a) and (b). In fact, in Fig. 8(a), the white symbols (corresponding to $5 \leq J \leq 8$) label lower efficiencies than those labeled by the black symbols (corresponding to $3 \leq J \leq 4$), which is the opposite of Fig. 8(b).

Therefore, given the parameters of a particular problem or application [i.e., a point $(J', L') \in \Sigma$], the designer can choose that architecture (between $\text{Arc}_J$ and $\text{Arc}_L$) that is the most efficient in $(J', L')$. We call this architecture the *optimal architecture* for $(J', L')$, and we denote it as $\text{Arc}_{\text{OPT}}(J', L')$, which is simply given by

$$
\text{Arc}_{\text{OPT}} \equiv \begin{cases} 
\text{Arc}_J, & \text{if } \text{eff}[^{\text{Arc}_J}] \geq \text{eff}[^{\text{Arc}_L}] \\
\text{Arc}_L, & \text{if } \text{eff}[^{\text{Arc}_J}] < \text{eff}[^{\text{Arc}_L}]
\end{cases}
\quad \text{i.e., inequality (7) is not satisfied by } J' \text{ and } L'.
$$

Fig. 8(c) shows $\text{eff}[^{\text{Arc}_{\text{OPT}}}]$ in $\Sigma'$. The average value of $\text{eff}[^{\text{Arc}_{\text{OPT}}}]$ in $\Sigma'$ is 99.1%, and its minimum value is 93.8%.
VI. CONCLUSION

In this paper, we have proposed two scalable architectures (ArcJ and Arc2J) that perform a DWT of an N0/2-sample sequence in only N0/2 computations. Therefore, the proposed devices are at least twice as fast as the other known architectures. Moreover, they have an AT2 parameter that is approximately 1/2 that of already existing devices.

This result is twofold. First, ArcJ and Arc2J can be employed for performing two times faster processing than that allowed by other architectures working at the same clock frequency (high-speed utilization). Second, they can be employed, even using a two times lower clock frequency, but reaching the same performance of other architectures. This second possibility allows for reducing the supply voltage and the power dissipation, respectively, by a factor of two and four with respect to the other architectures (low-power utilization). These results have been achieved by means of pipelining. Even though other architectures have already exploited the pipelined approach, they do not reach the performance of the proposed architectures, and they result in heavy underutilization. In fact, the stage implementing the decomposition 13 level l in pipelined architectures is usually clocked at a frequency 2k−1 times lower than the clock used in the stage performing decomposition level 1. Conversely, our architectures have been designed taking into account the balancing of the pipeline. Therefore, they are highly efficient. Specifically, we have shown that the efficiency of ArcJ decreases with the number of levels J, while the efficiency of Arc2J grows with J. Therefore, as a conclusive result, an impressively efficient architecture has been defined [say, ArcOPT(J', L')] which is simply the architecture between ArcJ and Arc2J having the highest efficiency when it implements an L'-tap filter based DWT in J' decomposition levels.

The efficiency of ArcOPT(J', L') is excellent. For instance, its average value [computed in very wide set of points (J', L')] is 99.1%. Its minimum value is 93.8%, and, except for five points, the efficiency is not lower than 96.9%.

ACKNOWLEDGMENT

D. Guevorkian would like to express his deepest gratitude to his colleagues from the Signal Processing Laboratory, Tampere University of Technology, where he has been working on this paper.

REFERENCES

Francescomaria Marino was born in 1968. He received the Laurea degree (cum laude) and the Ph.D. degree in electronic engineering from Polytechnic of Bari, Italy, in 1991 and 1996, respectively. Since 1999, he has been Assistant Professor in the Department of Electrical and Electronic Engineering, Polytechnic of Bari. In the same year, he became an Invited Visiting Researcher in the Signal Processing Laboratory, Tampere University of Technology. Previously, he worked in the Telecommunications Research Center of the Arizona State University (1998), in the Electrical and Computer Engineering Department of the University of Texas at Austin (1997), and in the Institute of Signal and Image Processing of the Italian National Council of Researches (1996). He has coauthored two patents by Intel and one patent by CNR. His main interests are parallel algorithms and architectures for digital image and signal processing. Dr. Marino received an award from Firestone S.p.A. as the Best Graduate of the Year in Universities of Puglia and an award from Telecom for his thesis work.

David Guevorkian received the Candidate of Sciences degree in physics and mathematics from Kiev State University, Ukraine, the M.Sc. degree in mathematics from Yerevan State University, Armenia, and the Dr.Tech. degree in signal processing from Tampere University of Technology, Finland, in 1997, where he is pursuing the Ph.D. degree. He was a Senior Researcher with the Institute for Problems of Informatics and Automation, National Academy of Sciences of Armenia, from 1983 to 1994. He then joined the Signal Processing Laboratory of Tampere University of Technology, where he worked until March 2000. Currently he is with Nokia Research Center, Finland. His research interests include signal and image processing, parallel algorithms and architectures, and computational complexity analysis.

Jaakko T. Astola was born in Helsinki, Finland, on May 6, 1949. He received the B.Sc., M.Sc., Licentiate, and Ph.D. degrees in mathematics (specializing in error-correcting codes) from Turku University, Finland, in 1972, 1973, 1975, and 1978, respectively. From 1976 to 1977 he was a Research Assistant at the Research Institute for Mathematical Sciences, Kyoto University, Kyoto, Japan. Between 1979 and 1987, he was with the Department of Information Technology, Lappeenranta University of Technology, Lappeenranta, Finland, holding various teaching positions in mathematics, applied mathematics, and computer science. In 1984, he was a Visiting Scientist at Eindhoven University of Technology, the Netherlands. From 1987 to 1992, he was an Associate Professor in applied mathematics at Tampere University, Tampere, Finland. Currently, he is a Professor of signal processing and Head of the Signal Processing Laboratory of Tampere University of Technology, elected by the Finnish Ministry of Education as a Center of Excellence in Research in Finland. He leads a group of about 50 scientists.