

Designing Low Power Array Architectures Based on Reversible Pipeline Method¹

D. Soudris, C. Z. Lolas, and A. Thanailakis

VLSI Design and Testing Center

Laboratory of Electrical and Electronic Materials Technology

Department of Electrical and Computer Engineering

Democritus University of Thrace, 67 100 Xanthi, Greece

Abstract : A new method for designing low power architectures based on the adiabatic switching techniques is introduced. The proposed architectures embody the principles of the reversible pipeline approach, which can ensure the adiabatic function of a circuit. Exploiting the inherent property of a reversible pipeline system that the “reverse” logic blocks are used in different time instances and the characteristic that several real life applications have identical logic blocks, we can reduce the hardware complexity significantly. More specifically, an appropriate multiplexer component and suitable clocking scheme achieve the reversible operation with small hardware cost. A series of proven lemmas specify the optimal characteristics of the chosen clocks, which ensure adiabatic operation and high speed. The proposed technique is suitable for designing array processor architectures, which implement a certain class of Digital Signal Processing applications.

Keywords: low power design, adiabatic technique, reversible pipeline, hardware reduction, array processor, DSP algorithm
CSCC'99 Proceedings: - Pages 4111-4116

1. INTRODUCTION

Low power dissipation has become a significant parameter for implementing efficient digital systems. The market demands for portable devices with high performance forced the designers to invent methods for reducing power dissipation in VLSI systems. Several methods have been published and can be divided into two main categories: i) the conventional low power design approaches [1] and ii) non-conventional [2]. More specifically, the energy recovery is a non-conventional low power design approach, which is based on adiabatic switching principles [2]. Using adiabatic techniques, the signal transfer between circuit capacitances should be sufficiently slow, which implies that the energy dissipated as heat may be asymptotically low, during transfer. Moreover, the energy recovery systems can recycle back to a power source, the remaining energy stored on circuit capacitances.

In this paper, a new methodology for designing low power architectures based on the adiabatic switching techniques is introduced. The

derived architectures are characterized by reversible pipeline operation and small hardware complexity. A typical reversible pipeline system consists of a series of “forward” logic blocks and a series of the corresponding “reverse” pipeline logic blocks. Many practical applications, for instance, in Digital Signal Processing field can be implemented, applying conventional design methodologies, by regular, modular, and iterative architectures [6, 7]. Thus, the final architecture includes many identical blocks (or processing elements). Additionally, exploiting the inherent property of a reversible pipeline system that the “reverse” logic blocks are used in different time instances, we can design efficient reversible pipeline architectures. Therefore, it is possible to merge the identical parallel blocks to one employing appropriate space multiplexing of the common block. Apparently, the new clocking strategy of the multiplexed blocks should preserve the reversible pipeline principles. For that purpose, a systematic methodology to determine the optimal power clock strategy and

¹ This work was partially supported by the LPGD 25256 ESPRIT project in the context of Low Power Action of EC.

to reduce the overall hardware complexity is developed. A series of formally proven lemmas specify the optimal characteristics of the chosen clocks, which ensure adiabatic operation and high speed. The proposed technique is suitable for designing array processor architectures, which implement a certain class of Digital Signal Processing algorithms named as Weak Single Assignment Codes (WSACs) [7,8]. The effectiveness of the proposed method is illustrated by the architecture level design of the convolution algorithm.

2. THE PROPOSED METHODOLOGY

2.1 The Basic Design Idea

The Reversible Pipeline principle was described in detail manner by Athas et al [Ath94]. The general scheme of a Reversible Pipeline system and the clock scheme are depicted in Fig. 1 and 2. Concerning the Reversible Pipeline operation, the role of the reverse blocks is to recover circuit energy back to power supply and do not contribute to calculation process itself. In other words, their only usage is to establish a path to recover the energy back to the power source. Thus, if a forward logic block is removed from the pipeline, the system logic operation will be changed. In contrary, if a reversible logic block is removed, the calculation sequence will not be influenced. If some of the reverse blocks are identical, alternative energy paths are available in a reversible pipeline system. By definition, a reversible pipeline system uses the reverse logic blocks in different time instances as the calculation and pipeline proceeds from a logic block, F_m to the next F_k one. Therefore, the hardware utilization of the reverse blocks is very small, i.e. $1/N$ %. To improve the hardware utilization, which coincides with the hardware reduction, we can merge appropriate groups of reverse logic blocks to one block. Since the hardware-reduced system should preserve the reversible pipeline principles (i.e. energy recovery), an component, which multiplexes in-time the paths of the paths of reverse blocks and manipulates the associated clocks of the merged blocks in effective fashion, should be used. For example, it may be a multiplexer. The derived

system can function as a pipeline system with less hardware complexity. A systematic methodology for designing the structure of the multiplexer and the characteristics of the multiplexed clocks will be described in the following sections.

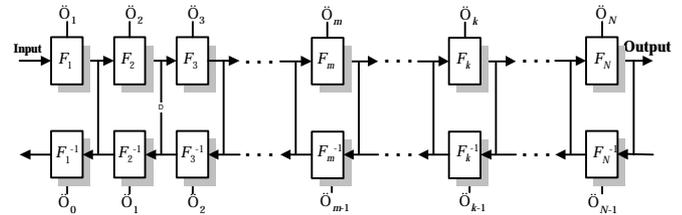


Figure 1. The general structure of a reversible pipeline architecture [2]

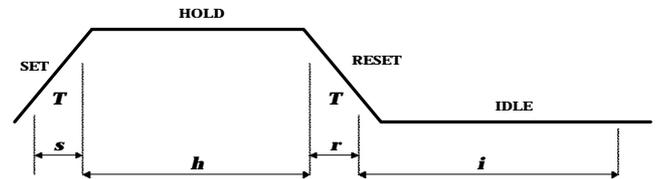


Figure 2. The clock characteristics of a reversible pipeline system.

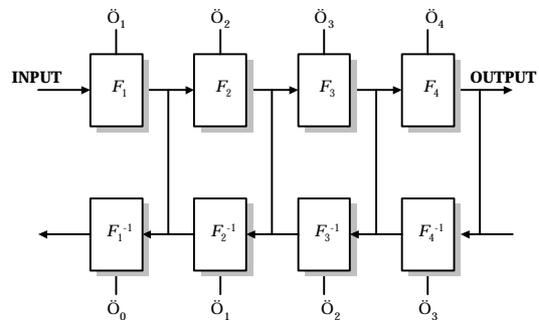


Figure 3. The existing reversible pipeline architecture with $N=4$.

For simplicity reasons, the function of the derived architecture will be explained using a certain reversible pipeline system with four stages ($N=4$) shown in Fig. 3. This system has two identical reverse logic blocks,

the second and the fourth one. Fig. 4 depicts the proposed architecture.

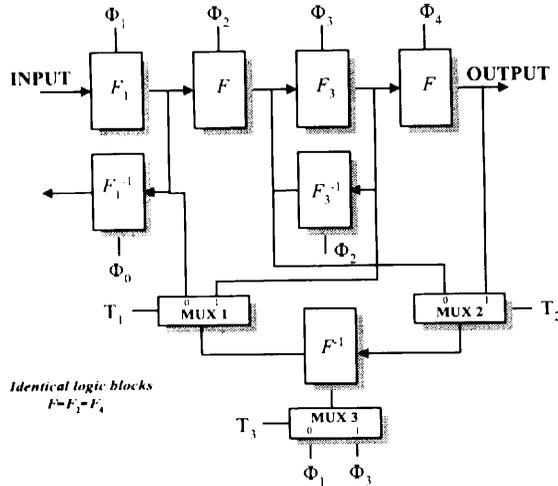


Figure 4. The optimized reversible pipeline architecture (N=4).

2.2 METHOD FOR SELECTING OPTIMAL CLOCKING SCHEME

For an N -stage pipeline (F_1, F_2, \dots, F_N) the clocks are being used are $\Phi_1, \Phi_2, \dots, \Phi_N$. From now on a block which is picking by chance will be described as F_m or F_k .

First design criterion: the “shared” reverse block should be used in different time intervals.

Second design criterion: the clocks of the «shared» reverse blocks must have non-interference pulses

In the following paragraphs a set of definitions and lemmas are given. These lemmas facilitate the extraction of fast and secure solutions about the appropriate clocking scheme for the optimization technique. This procedure is only valid for a pipeline system in which the identical reverse blocks are being merged in couples.

In the beginning, is it necessary to define some variables, which will help the formal description of the proposed clocking scheme. Referring to Fig. 2, we define:

- i. T = raising/falling time of a clock
 - ii. SET = time interval with length s ,
 - iii. HOLD = time interval with length h ,
 - iv. RESET = time interval with length r ,
 - v. IDLE = time interval with length i ,
- and

- vi. DIFFERENCE=time difference interval between two clocks with length d . Since the raising/falling edges occur during the SET/RESET intervals, we deduce that:

$$s = r = T \quad (1)$$

Without loss of generality, it is assumed that the lengths of HOLD, IDLE, and DIFFERENCE intervals can be expressed in terms of T . That is:

$$h = x T \quad (2a)$$

$$i = y T \quad (2b)$$

$$d = z T \quad (2c)$$

where $x, y \in \mathbf{R}^+ - \{0\}$, and (\mathbf{R}^+ is the positive of real numbers). In general, any two clocks used in reversible pipelined systems are identical and have equal phase difference. These features arise from the fact that the clock implementation is easier.

Corollary 1: Due to the fact that the difference between two successive blocks is assumed to be constant, d , the time shift between two clocks Φ_m and Φ_k of the stages F_m and F_k is equal to:

$$(m - k) z T \quad (3)$$

where $m > k$ with $m, k \in \mathbf{Z}^+ - \{0, 1\}$ (\mathbf{Z} is the set of positive integer numbers).

Lemma 1: Let s, h, r and i be a set of the time intervals of a clock period as they have been defined previously. The relationship between the first three intervals and last one is:

$$2 + x \leq y \quad (4)$$

Lemma 2: Two successive clock signals are partially overlapped if it holds:

$$1 \leq z < x \quad (5)$$

Lemma 3: Let F_m and F_k be two identical reverse blocks and Φ_m and Φ_k be the associated clocks, which meet the second criterion. The phase difference between Φ_m and Φ_k should meet the formula:

$$(m - k) z \geq 2 + x \quad (6)$$

Lemma 4: Let F_m and F_k be two identical reverse blocks and Φ_m and Φ_k be associated clocks, which meet the second criterion. The phase difference between Φ_m and Φ_k should meet the formula:

$$(m - k) z \leq y \quad (7)$$

Given the values m and k of the identical logic blocks and (4), (6) and (7), we

can determine a clocking scheme choosing one of the possible combinations of x , y , and z , which satisfy the inequalities:

$$2 + x \leq (m - k) z \leq y \quad (8)$$

$$1 \leq z < x \quad (9)$$

Lemma 5. Given the inequalities (8) and (9), it holds that:

$$i) \quad y \geq 2 \quad (10)$$

$$ii) \quad (m - k) > 1 \quad (11)$$

The clocks deriving from the above inequalities are suitable for merging every pair of identical blocks, which has the same relative position in the pipeline stream, i.e. constant difference $(m - k)$ between F_m and F_k . This important result has impact on the design of systems, which are characterized by iterative (repeated) identical structures. Such typical systems are the array processors [6, 7], that can implement several of Digital Signal Processing (DSP) applications. Therefore, the proposed method can be applied for implementing DSP applications in architecture level, which embody the properties of the reversible pipeline.

2.3 ARCHITECTURE CHARACTERISTICS

2.4.1 Energy dissipation

If the proposed system has the appropriate design and clocking strategy, then the power consumption is not influenced by the proposed technique. That is, the energy savings of the optimized reversible pipeline system are equal to original one.

2.4.2 System speed

Apparently, the operation frequency of a reversible pipeline depends on the period and the phase difference of the used partially overlapped clocks. On the other hand, the clocking strategy of a hardware-reduced RP system design is subject to a series of criteria, which may restrict the designer to reach optimal solutions. The clocks resulting from (8) and (9) do not lead necessarily to maximum speed or at least equal to the speed of a conventional RP system. Consequently, the designer should examine how can maximize the speed of the whole system. It has been already mentioned that the factors, which influence the features of a clock pulse are T , x , y , and z . More specifically, in circuit level,

the speed depends on the rising/falling time T . In system level, the speed depends on the time interval, t_1 , needed to obtain the first output (i.e. latency) and the time interval, t_2 , in which a logic block can load new inputs after the execution of a computation (i.e. throughput). The time interval t_1 and t_2 can be expressed in terms of T , x , y , and z as follows:

$$t_1 = a z T \quad (12)$$

$$t_2 = (2+x+y)T \quad (13)$$

where a is the number of the forward logic blocks. Therefore, from all the possible solutions of (8) and (9), the set of x , y , and z , which minimizes eq. (12) and (13) ensures maximum speed.

2.4 HARDWARE COMPLEXITY

In order to estimate the total hardware complexity of the proposed pipeline implementation, it is necessary to examine the hardware complexity of the additional components, i.e. the multiplexers. We select a fully-adiabatic multiplexer, which is based on a fully adiabatic tree decoder designed by T-gate logic [Ath94]. A simple 2-to-1 adiabatic multiplexer consisting of 2 input lines and 1 output line and implemented by 4 transistors (2 p-MOS and 2 n-MOS). Generally, the implementation of a $2M$ -to- M multiplexer requires $4M$ transistors. In general, a qM -to- M multiplexer requires $2qM$ transistors

In order to estimate the hardware savings of the proposed system, we define the total *Relative Hardware Saving* (RHS_T) as follows:

$$RHS_T = \frac{H_B - H_A}{H_B} \quad (14)$$

where H_B is the total amount of hardware (in number of transistors) of the reverse blocks before merging and H_A is the total amount of hardware (in number of transistors) of the reverse blocks after merging.

The RHS_T can be expressed by a more specific formula, if all the blocks of the reversible pipeline system are identical, for instance, the array processor architectures (systolic architectures) [6,7]. More specifically, given that, $H_B = A N_T$ and $H_A =$ number of remaining reverse

blocks* N_T + number of transistors in MUXs, eq. (14) becomes:

$$RHS_T = 1 - \frac{1}{g} - \frac{2(M + N + 1)}{N_T} \quad (15)$$

where:

A = the number of reverse blocks of the whole system,

N_T = the number of transistors of each block,

g = the number of blocks each group consists of (permissible values are 2, 4, 8, 16,), and

C = the total number of groups.

A significant conclusion comes from the previous expression: if all the blocks are identical, RHS_T is expressed in terms of a block characteristics and the number of the identical blocks of each group, and it is independent from of the total number of the reverse blocks. The RHS_T is becoming better as g is increasing. However, if g increases, $m-k$ becomes larger as well as the values x , y , and z of (8) and (9). This fact implies increased latency, t_1 , and thus lower speed. Depending on the application requirements, the designer should make the appropriate trade-offs between RHS_T and speed, choosing the suitable value of g .

3. APPLICATION

The features and the advantages of the proposed modified reversible-pipelining approach are illustrated by the well-known DSP algorithm of convolution. It has been proven that the convolution algorithm can be implemented by regular array architectures [6, 7]. This system is partially adiabatic and it is designed to recover the external signal energy, that is the energy of the input/output signals of the processing elements.

For comparison reasons, we choose a specific array processor, which implement the convolution of two sequences of eight ($A=8$) points, with certain 32×32-bit multiplier-accumulator unit (i.e. processing element) of 28,500 transistors [Mur96]. Hence, from eq. (15), we infer that:

$$RHT_T = 0,986 - \frac{1}{g} \quad (16)$$

There are two different ways for merging identical the blocks of the Signal Flow Graph, that is: $g=2$ and $g=4$. For arbitrary length of the convolved sequences, Fig. 5 depicts the relationship between RHS_T and the number of blocks of each group, g .

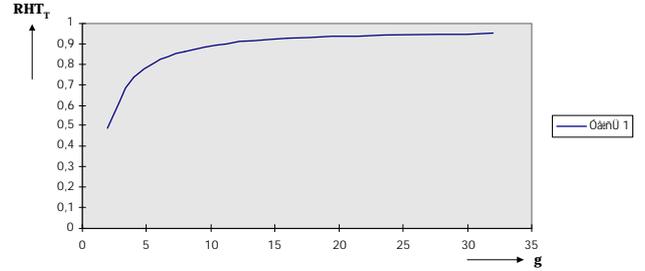


Figure 5. The relationship between RHS_T and the number of blocks of each group, g .

Groups of two

There are two ways for merging the reverse blocks of this system: the first is to merge blocks by pairs having $m-k = 2$ (i.e. blocks 1-3, 2-4, 5-7, 6-8) and the second is to merge blocks with $m-k = 4$ (i.e. blocks 1-5, 2-6, 3-7, 4-8). From the proposed method, choosing $m-k = 2$, we result into an architecture (Figure 6) with higher speed. The optimal clocking scheme, which results from (8) and (9), has $x = 2.2$, $y = 4.2$ and $z = 2.1$. The associated values of RHS_T , throughput and latency are in Table1.

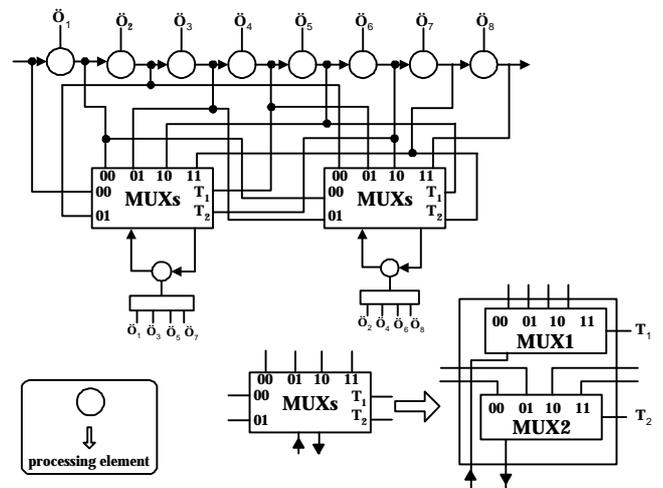


Figure 6. The reversible-pipeline array processor architecture of convolution algorithm of eight (8) points with $g=2$.

Groups of four

The merging of the eight reverse blocks with groups of four ($g=4$) has only one combination of identical blocks, that is, 1-3-5-7 and 2-4-6-8. The deriving system is shown in Fig. 7. Similarly, from (8) and (9) we infer that the minimum solution is satisfied by $x = 2.2$, $y = 12.6$ and $z = 2.1$. The values of RHS_T , throughput and latency are depicted in Table 1.

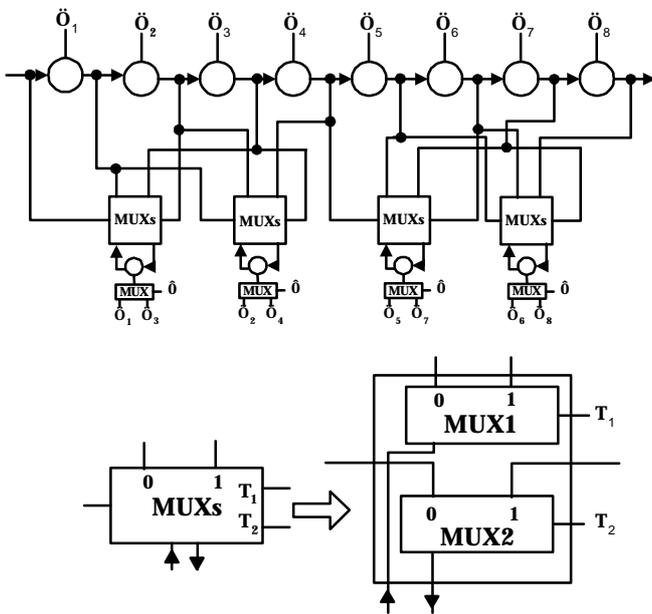


Figure 7. The reversible-pipeline array processor architecture of convolution algorithm of eight (8) points with $g=4$.

g	RHS_T (hardware reduction)	Throughput (t_1)	Latency (t_2)
2	48.6%	16.8 T	8.4 T
4	73.6%	16.8 T	14.8 T

Table 1. The hardware reduction and timing requirements of two linear array processors.

In conclusion, the merge as many as possible identical blocks results into larger hardware reduction at the expense of architecture latency.

4. CONCLUSIONS

A new systematic methodology for implementing low power architectures, which are based on adiabatic techniques was presented. The derived architectures exploit the properties and features of the reversible pipelining and exhibit significantly-reduced hardware complexity. A series of proven lemmas determine the exact and optimal characteristics of the used clocks. The efficacy of the proposed architectures was illustrated by a certain DSP algorithm (convolution).

REFERENCES

- [1] J. Rabaey and M. Pedram, "Low Power Design Methodologies," Kluwer Academic Publishers, 1996.
- [2] W.C. Athas, L. Svensson, J.G. Koller, N. Tzartzanis, E.Y.C. Chou, "Low-Power Digital Systems Based on Adiabatic-Switching Principles," in IEEE Trans. on VLSI Systems, vol. 2, no.4, Dec. 1994.
- [3] H. Murakami, et al, "A multiplier accumulator macro for a 45 MIPS embedded RISC processor," in IEEE J. Solid State Circuits, July 1996, 31, pp. 1067-1071.
- [4] K. Jung and W. Kim, "A Logic gate for Reversible Pipelining," in Proc. of IEEE Int. Work. on LPD, 1997, pp. 1928-1931.
- [5] S.G. Younis and T.F. Knight, Jr., "A asymptotically zero energy split-level charge recovery logic," in Proc. of IEEE Int. Work. on LPD, 1994, Napa-Valley, USA.
- [6] S.Y. Kung, "VLSI Array Processors," Prentice Hall, Eaglewood Cliffs, 1988.
- [7] D. Soudris et al, "On the Design of Two-Level Pipelined Processor Arrays," in Application-Driven Architecture Synthesis, F. Catthoor and L. Svensson editors, Kluwer Academic Publishers, June 1993.
- [8] V. Roychowdhury, et al, "On the Localization of algorithms for VLSI

processor arrays," in *VLSI Signal Processing III*, edited by R. Broadersen and H. Moscovitz, IEEE Press, New York, pp. 459-470, 1988.