Provided for non-commercial research and education use. Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/authorsrights

Operations Research Letters 41 (2013) 486-489

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

On optimal flip-flop grouping for VLSI power minimization

Shmuel Wimer*

Bar-Ilan University, Engineering Faculty, Ramat-Gan, Israel Technion-Israel Institute of Technology, EE Department, Haifa, Israel

ARTICLE INFO

Article history: Received 13 December 2012 Received in revised form 7 June 2013 Accepted 7 June 2013 Available online 14 June 2013

Keywords: NP-hardness Minimum cost perfect matching VLSI clock gating VLSI power optimization

ABSTRACT

Data-driven clock gating is reducing the total power consumption of VLSI chips by 20%. There, flip-flops are grouped and share a common clock signal. Finding the optimal clusters is the key for maximizing the power savings. Clustering by the minimal cost perfect graph matching algorithm (MCPM) proposed by other works is not optimal. We show that the optimal clustering problem is NP-hard, and study the quality of MCPM heuristics, showing by experiments that it falls 5% above the optimal solution. © 2013 Elsevier B.V. All rights reserved.

1. Introduction

One of the major switching power consumers in computing and consumer electronics products is the system's clock signal, typically responsible for 30% to 70% of the total switching power consumption [11]. Several techniques to reduce the switching power have been developed, of which clock gating is predominant. Ordinarily, when a logic unit is clocked, its underlying sequential elements (flip-flops) receive the clock signal regardless of whether or not they will toggle in the next cycle. With clock gating, the clock signals are conditioned (ANDed) with explicitly predefined enabling signals. Clock gating is employed at all chip levels: system architecture, block design, logic design and gates [2,8]. Several methods to take advantage of this technique are described in [3,6,13], with all of them relying on various heuristics in an attempt to increase clock gating opportunities.

This paper studies *data-driven clock gating*, employed for flipflops (FFs) at the gate-level, which is the most aggressive possible. The clock signal driving an FF is disabled (gated) when the FF's state is not subject to change in the next clock cycle [5]. In a recent study, a model for data-driven gating was developed based on the toggling (switching) activity (state changes) of the constituent FFs [15]. The optimal fan-out of a clock gater yielding maximal power savings was derived based on the average toggling statistics of the individual FFs. While [15] answered the question of what is

0167-6377/\$ - see front matter © 2013 Elsevier B.V. All rights reserved. http://dx.doi.org/10.1016/j.orl.2013.06.002 the group size that maximizes power savings, this work studies the problem of which FFs should be placed in a group to maximize the power reduction and how to algorithmically derive those groups. We subsequently use the terms activity, toggling and switching interchangeably.

A data-driven clock gating circuit is illustrated in Fig. 1. By XORing its output with the present input data that will appear at its output in the next clock cycle, an FF checks whether its state is subject to change, thus finding out whether its clock can be disabled in the next cycle. The outputs of k XOR gates are ORed and then latched to generate a joint gating signal for k FFs. The combination of a latch with AND gate is called Integrated Clock Gate (ICG) [10], commonly used by commercial electronic design automation (EDA) tools. Notice that a single ICG is amortized over k FFs. There is a clear tradeoff between the number of saved (disabled) clock pulses and the hardware overhead. With an increase in k the hardware overhead decreases, but so does the probability of disabling, obtained by ORing the k enable signals.

The FFs of a system need to be clustered in k-size sets such that the power savings will be maximized. Grouping FFs for joint clock gating was described in [12] as a part of physical layout synthesis. While addressing design factors as clock-skew, power, and area minimization, it was not aware of the toggling correlations of the underlying FFs, which this paper does. The optimal value of kwas obtained by [15] under worst-case assumption of FFs toggling independence. In reality however, the toggling is correlated, so one can expect for higher savings than the theoretical lower bound obtained under independence assumption. In the sequel we use the terms grouping and clustering interchangeably.

The next section presents the problem of optimal FF grouping by introducing a graph model followed by a formulation of the





Operations Research



^{*} Correspondence to: Bar-Ilan University, Engineering Faculty, Ramat-Gan, Israel. Tel.: +972 4 8295744.

E-mail addresses: wimer@ee.technion.ac.il, wimers@biu.ac.il.



Fig. 1. Data-driven clock circuit. Overhead hardware is shaded in gray.

associated optimization problem. Section 3 shows the inherent difficulty of the problem and proves its NP-hardness. Section 4 describes a heuristic grouping algorithm.

2. Optimal FFs grouping for joint clock gating

Let *n* FFs be clocked during m + 1 cycles. A first step towards an optimal FFs grouping is to take advantage of the correlations of their toggling. Let $\mathbf{a} = (a_1, \ldots, a_m)$ be the activity of an FF, where $a_t = 0, 1 \le t \le m$, if the FF stays unchanged (no toggling) from time t-1 to time t, and $a_t = 1$, otherwise. The term $\|\mathbf{a}\| = \sum_{t=1}^m a_t$ is proportional to the power consumed by the FF's switching. All the n (n - 1) / 2 pairs $(\mathbf{a}_i, \mathbf{a}_j), 1 \le i < j \le n$, are bit-wise XORed to yield the number $\|\mathbf{a}_i \oplus \mathbf{a}_j\|$ of *redundant clock pulses* occurring if FF_i and FF_j are jointly clocked by a common gater. A key consideration in selecting FFs to be driven by a common gater is their activity similarity given by $\|\mathbf{a}_i \oplus \mathbf{a}_j\|$. The smaller it is, the more desirable it is to jointly clock FF_i and FF_j.

To model the switching power consumed when driving FFs pairs (k = 2) with a common clock gater, an *n*-vertex complete weighted graph G(V, E, w), called *FF pairwise activity graph*, is defined. Assume w.l.o.g that *n* is even (we could otherwise add a never toggling artificial FF and set to zero the weight of its entire incident edges). A vertex $v_i \in V$ is associated with FF_i's activity \mathbf{a}_i . An edge $e_{ij} = (v_i, v_j) \in E$ is associated with a joint activity vector $\mathbf{a}_i | \mathbf{a}_j$, where the OR is a bit-wise operation. An edge e_{ij} is assigned a weight $w(e_{ij}) = \|\mathbf{a}_i \oplus \mathbf{a}_j\|$, counting the number of redundant clock pulses incurred by clocking FF_i and FF_j with a common gater. Let $E' \subset E$, |E'| = n/2, be a vertex matching of G(V, E, w). The total power *P* consumed by the clock signal depends on the number of clock pulses driving the underlying FFs, given by

$$P = 2 \sum_{e_{ij} \in E'} \|\mathbf{a}_i\| \mathbf{a}_j\|$$

= $\sum_{v_i \in V} \|\mathbf{a}_i\| + \sum_{e_{ij} \in E'} [\|\mathbf{a}_i \oplus (\mathbf{a}_i|\mathbf{a}_j)\| + \|\mathbf{a}_j \oplus (\mathbf{a}_i|\mathbf{a}_j)\|]$
= $\sum_{v_i \in V} \|\mathbf{a}_i\| + \sum_{e_{ij} \in E'} \|\mathbf{a}_i \oplus \mathbf{a}_j\| = \sum_{v_i \in V} \|\mathbf{a}_i\| + \sum_{e_{ij} \in E'} w(e_{ij}).$ (1)

The first sum on the right hand side of (1) is an essential power component charged to the toggling of the individual FFs, and is independent of the pairing. Therefore, to consume minimum switching power (or alternatively, achieve maximum power savings) it is necessary to minimize $\sum_{e_{ij} \in E'} w(e_{ij})$, which turns into the well-known minimal cost perfect graph matching (MCPM) problem, for which polynomial complexity algorithms are known [9].

The extension for k > 2 is straightforward. Assume w.l.o.g that *n* is divisible by *k*. (We could otherwise add a never toggling

artificial FFs and set to zero the weight of their entire incident edges.) A complete *k*-uniform weighted hypergraph H(V, E, w), called *FF grouping activity hypergraph*, is defined, where for a subset $\mathbf{v} \subset V$ and $|\mathbf{v}| = k$, $e_{\mathbf{v}} = \{v_u\}_{u \in \mathbf{v}} \in E$ defines a hyper edge. It follows that $|E| = \binom{n}{k}$. A hyper edge $e_{\mathbf{v}}$ is associated with a joint activity vector $\bigcup_{u \in \mathbf{v}} \mathbf{a}_u$, defined by the bit-wise ORing of the *k* toggling vectors. A hyper edge $e_{\mathbf{v}}$ is assigned a weight

$$w\left(\boldsymbol{e}_{\mathbf{v}}\right) = \sum_{v \in \mathbf{v}} \left\| \mathbf{a}_{v} \oplus \bigcup_{u \in \mathbf{v}} \mathbf{a}_{u} \right\|,\tag{2}$$

which is the total number of redundant clock pulses incurred by clocking the k FFs corresponding to e_v with a common gater.

Let $E' \subset E$ be an exact cover of the vertices of H(V, E, w) by n/k hyper edges (a vertex belongs to one and only one hyper edge). The total power *P* consumed by the clock signal depends on the total number of pulses driving the FFs, given by

Ш

Ш

$$P = \sum_{e_{\mathbf{v}} \in E'} k \left\| \bigcup_{u \in \mathbf{v}} \mathbf{a}_{u} \right\|$$

$$= \sum_{v_{i} \in V} \|\mathbf{a}_{i}\| + \sum_{e_{\mathbf{v}} \in E'} \sum_{v \in \mathbf{v}} \left\| \mathbf{a}_{v} \oplus \bigcup_{u \in \mathbf{v}} \mathbf{a}_{u} \right\|$$

$$= \sum_{v_{i} \in V} \|\mathbf{a}_{i}\| + \sum_{e_{\mathbf{v}} \in E'} w (e_{\mathbf{v}}).$$
(3)

The first sum on the right hand side of (3) is an essential power component charged to the toggling of the individual FFs, and is independent of the grouping. Therefore, to consume minimum switching power (or alternatively, achieve maximum switching power savings) it is necessary to minimize $\sum_{e_v \in E'} w(e_v)$, a problem we call *MIN_CLK_GATE*. With the above formulation, a solution to the problem of finding n/k hyper edges exactly covering the n vertices and yielding minimum redundant clock pulses can be obtained by solving the well-known NP-hard weighted *Set Partitioning Problem* (SPP) [1]. There, hyper edges are the variables covering the vertex constraints.

Several papers have addressed the grouping problem based on FFs pairing (k = 2), and applied the solution iteratively to larger sets where $k = 2^{K}$. The work in [4] proposes heuristics where FFs are sorted by their activity and then paired in that order. It is possible however, to construct an example where this heuristic would increase the number of redundant clock pulses rather than reduce them. In [13,14], FFs were paired based on intuitive arguments without a formal proof, and it may sometimes yield inferior gating. The work in [6] proposed a bottom-up process by repeating the MCPM algorithm. Starting with *n* individual FFs and constructing the associated *n*-vertex FF pairwise activity graph, an MCPM algorithm then finds the best FFs pairing. A new *n*/2-vertex pairwise activity graph is then defined where its vertices correspond to the *n*/2 edges of the matching found in the former step. The process repeats *K* times until groups of size $k = 2^{K}$ are determined.

3. NP-hardness of optimal FF grouping

For k = 2 (K = 1) MCPM indeed solves the problem of minimizing the number of redundant clock pulses, but its repetitive application for k > 2 (K > 1), as was proposed in [6], may not find the minimum. This is demonstrated by the counter example illustrated in Fig. 2, where k = 4 (K = 2). The toggling vectors of eight FFs are shown in Fig. 2(a). Applying MCPM yields the pairs (FF₁, FF₂), (FF₃, FF₄), (FF₅, FF₆) and (FF₇, FF₈) with $||\mathbf{a}_1 \oplus \mathbf{a}_2|| + ||\mathbf{a}_3 \oplus \mathbf{a}_4|| +$ $||\mathbf{a}_5 \oplus \mathbf{a}_6|| + ||\mathbf{a}_7 \oplus \mathbf{a}_8|| = 15$ redundant clock pulses (the red Os in Fig. 2). This is indeed the optimal pairing of FFs (k = 2). S. Wimer / Operations Research Letters 41 (2013) 486-489

FF1:	0	<mark>0</mark>	1	0	0	<mark>0</mark>	1	0	<mark>0</mark>	<mark>0</mark>	0	1	
FF2:	0	1	0	0	0	1	1	0	1	1	0	1	
FF3:	1	1	1	0	0	0	0	1	<mark>0</mark>	1	1	0	
FF4:	1	0	1	0	0	0	0	1	1	1	1	0	
FF5:	1	0	0	1	1	0	1	0	1	1	0	<mark>0</mark>	a
FF6:	1	0	1	1	1	0	1	0	1	0	0	1	
FF7:	0	1	1	1	<mark>0</mark>	1	0	<mark>0</mark>	1	0	0	1	
FF8:	0	1	1	1	1	0	0	1	0	0	0	0	
FF1:	0	0	1	0	0	0	1	0	0	0	0	1	
FF2:	0	1	0	0	0	1	1	0	1	1	0	1	
FF6:	1	0	1	1	1	0	1	0	1	0	0	1	
FF7:	0	1	1	1	0	1	0	0	1	0	0	1	
FF3:	1	1	1	0	0	0	0	1	0	1	1	0	b
FF5: FF8:	1 0	0 1	0 1	1 1	1 1	0	1 0	0 1	1 0	1 0	0 0	0	

Fig. 2. A counter example to 4-size FF grouping by bottom-up MCPM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

However, the optimal 4-size grouping is (FF_1, FF_2, FF_6, FF_7) and (FF_3, FF_4, FF_5, FF_8) , yielding 35 redundant clock pulses. The pairs (FF_5, FF_6) and (FF_7, FF_8) have been split between the two 4-size sets shown in Fig. 2(b). Consequently, the optimal solution could not be obtained by a repetitive MCPM. This follows from the inherent difficulty of the MIN_CLK_GATE problem, shown next to be NP-hard.

We prove the NP-hardness of MIN_CLK_GATE by a polynomial reduction of the X3C *exact covering problem* [7] into a decision problem version of MIN_CLK_GATE. Minimizing $\sum_{e_v \in E'} w(e_v)$ in (3) can equivalently be viewed as maximizing the number of clock pulses saved by FF grouping, where an FF that is not subject to toggling at the next clock cycle does not receive a clock pulse. Consider the following version of the decision problem:

Problem: MIN_CLK_GATE

Instance: A set of *n* FFs where *n* is divisible by *k*. Each FF is associated with a toggling vector $\mathbf{a} = (a_1, a_2, ..., a_m)$ during m + 1 clock cycles, where $a_j = 1$ if the FF changes states between clock cycles j - 1 and j, $1 \le j \le m$, and $a_j = 0$ otherwise.

Question. Is there a grouping of the *n* FFs in n/k *k*-size disjoint sets such that savings of at least *n* clock pulses are achieved by jointly driving the FFs in each set with a common clock gater?

Theorem. The MIN_CLK_GATE decision problem is NP-complete.

Proof. We will consider a special case of MIN_CLK_GATE where *n* is divisible by k = 3 and in every clock cycle exactly n - 3 FFs are toggling. The implied decision problem is whether there exists a grouping of the *n* FFs in 3-size sets, resulting in savings of *n* clock pulses at least. The toggling vectors of the FFs during the m + 1 clock cycles are represented by a $n \times m$ matrix $\mathbf{T} = (t_{ij})$ defined as follows:

$$t_{ij} = \begin{cases} 1 & \text{FF}_i \text{ is toggling from } j - 1 \text{ to } j \text{ clock cycle} \\ 0 & \text{otherwise,} \end{cases}$$
$$1 \le i \le n, \ 1 \le j \le m. \tag{4}$$

Let the 3-size set $\{FF_q, FF_r, FF_s\}$ be driven by a common clock gater, and let $T_{\{q,r,s\}}$ be its corresponding $3 \times m$ sub matrix of **T**. There can be any number, from zero to three, of 1s in a column of $T_{\{q,r,s\}}$. If all elements are 0, ORing the three 0s yields a 0 enabling signal and the gater will not send a clock pulse to any of $\{FF_q, FF_r, FF_s\}$, hence three pulses are saved. In all other case at least one of the three FFs toggles and must therefore be clocked. An ORing results in a joint enabling signal of 1 and all the three FFs will receive a clock pulse, thus no savings occur. Once the FFs are grouped in 3-size sets, calculating the implied savings is straightforward and takes O(mn) time, showing that MIN_CLK_GATE decision problem is NP.

We will show next a polynomial reduction of the X3C decision problem [7] into the MIN_CLK_GATE decision problem. Recall that a X3C asks whether for a given set **S** of subsets, **S** = { S_1, \ldots, S_m }, $|S_j| = 3, 1 \le j \le m$, of a universal set of elements $U = \{U_1, \ldots, U_n\}$, there is a subset $S' \subseteq S$ such that $\bigcup_{S_j \in S'} S_j = U$ and $S_i \bigcap S_j = \emptyset$ for any $S_i \in S'$ and $S_j \in S'$, $i \ne j$. We may restrict ourselves to instances of the X3C cover problem where the subsets in **S** are unique.

A transformation f(I) of a X3C instance into MIN_CLK_GATE is defined as follows. An element $U_i \in U$, $1 \le i \le n$, is associated with an FF, and a subset $S_j \in S$, $1 \le j \le m$ is associated with a clock cycle. A $n \times m$ zero–one toggling matrix $\mathbf{T} = (t_{ij})$ is thus defined by:

$$t_{ij} = \begin{cases} 0 & \text{if } U_i \in S_j \\ 1 & \text{otherwise,} \end{cases} \quad 1 \le i \le n, \ 1 \le j \le m.$$

$$(5)$$

Since $|S_j| = 3$, a column in **T** has exactly three 0s.

Let the answer to MIN_CLK_GATE be yes. There is a grouping S', |S'| = n/3, of the *n* FFs into disjoint sets of 3 FFs each, yielding savings of *n* clock pulses at least. Consider a set {FF_q, FF_r, FF_s} \in S' and its corresponding $3 \times m$ sub matrix $\mathbf{T}_{\{q,r,s\}} \subset \mathbf{T}$, representing their joint toggling during the m + 1 clock cycles. Let p (FF_q, FF_r, FF_s) denote the number of clock pulses saved in jointly clocking {FF_q, FF_r, FF_s} during the m + 1 clock cycles. The yes answer to MIN_CLK_GATE means that:

$$\sum_{s'} p\left(\mathsf{FF}_q, \mathsf{FF}_r, \mathsf{FF}_s\right) \ge n. \tag{6}$$

We subsequently show that in (6) only the equality can hold. Let \mathbf{T}^{j} , $1 \le j \le m$, denote a column of \mathbf{T} , and let $p(\mathbf{T}^{j})$ be the number of clock pulses saved in a transition from clock cycle j - 1 to j. By the construction in (5) \mathbf{T} has exactly three 0s in every column, and since in $\mathbf{S} = \{S_1, \ldots, S_m\}$ of the X3C problem all S_j are distinct, at most one column of $\mathbf{T}_{\{q,r,s\}}$ has three 0s. The MIN_CLK_GATE grouping \mathbf{S}' comprises n/3 3-size FFs sets, so there are at most n/3 columns of \mathbf{T} where all three 0s fall within a set in \mathbf{S}' . Consequently:

$$\sum_{j=1}^{m} p\left(\mathbf{T}^{j}\right) \le 3 \times n/3 = n.$$
(7)

It follows from inequalities (6) and (7) that a *yes* answer to MIN_CLK_GATE implies total savings of exactly *n* clock pulses. These savings are obtained from the n/3 FFs sets, which from the transformation in (5) uniquely defines n/3 disjoint sets $\{U_q, U_r, U_s\}$, hence yielding a *yes* answer to the X3C problem.

The converse direction where a *yes* answer to X3C implies a *yes* answer to MIN_CLK_GATE is straightforward. We associate every FF_i with an element $U_i \in U$, and its toggling in a transition from clock cycle j - 1 to j, $1 \le j \le m$, with the definition of (5). A *yes* answer to X3C implies n/3 FFs sets satisfying (6), hence yielding a *yes* answer to the MIN_CLK_GATE decision problem. \Box

4. Results of iterative MCPM heuristic

We have shown that MIN_CLK_GATE is NP-hard. The MCPM heuristic to solve MIN_CLK_GATE is still practical, yielding results close to the minimal cost that can be obtained by SPP solution, as demonstrated by the following experiment. Since the number $\binom{n}{k}$ of SPP variables explodes with the number *n* of FFs and the group size *k*, we could afford a comparison of only a small case

Table 1

Results of FF grouping for the 3D graphics accelerator obtained by MCPM heuristic for $n = 4.9 \times 10^3$, $m = 10^5$ and p = 0.05.

Group size $k = 2^{K}$	2	4	8	16	32	64	128
$\frac{1}{\sum_{x = r'} w(e_x) (\times 10^6) \text{ Eq. (3)}}$	1.79	3.28	4.26	5.39	6.91	8.95	11.5
$mn \left[1 - p - (1 - p)^{k}\right] (\times 10^{6})$	23.3	66.4	140	249	370	447	465
Run time (min)	88	130	158	173	183	189	190

of n = 94 FFs, taken from a real VLSI design. The FF toggling benchmark spans $m = 10^5$ clock cycles and has average toggling p = 0.0736. The group size k = 4 is appropriate for the given toggling probability [15], resulting in a minimum cost SPP with $\binom{94}{4}\cong 3.05 imes 10^6$ variables and 94 constraints. Its solution was compared to the results obtained by MCPM heuristic. (Recall that for k = 2 both MCPM and SPP yield by their definition the same minimum.) The absolute minimum obtained by the minimum cost SPP algorithm has $\sum_{e_{\mathbf{v}} \in E'} w(e_{\mathbf{v}}) = 578,671$ redundant pulses, while the MCPM algorithm yields 604, 545 redundant pulses, which is 4.47% above the optimal solution.

The MCPM algorithm has also reasonable run time performance as shown in Table 1, obtained from a design comprising n = 4.9×10^3 FFs. The toggling benchmark spans $m = 10^5$ clock cycles and has p = 0.05 average FF toggling. The experiment ran on a 2 GHz processor with 2 GB RAM. Since all FFs pairs are allowed, the pairwise activity_graph includes 4.9 \times 10^3 vertices and $n(n-1)/2 \cong 1.2 \times 10^7$ edges. Due to FFs placement and proximity constraints resulting from VLSI design considerations (whose discussion is beyond the scope of this paper), the size of such a graph in practice is much smaller. Groups of $k = 2^{k}$ 2, 4, 8, ..., 128 have been examined. The results are summarized in Table 1. There, the number of redundant clock pulses as obtained by (3) is far smaller than that implied by a random grouping, yielding $mn \left[1 - p - (1 - p)^k\right]$ redundant pulses. The low number of redundant pulses obtained by the MCPM heuristic compared to random grouping stems from the correlations of FFs' activities which the grouping algorithm took advantage of. The run-time growth is nearly logarithmic in K. This follows from the iterative nature of group constructions where at each step a problem half the size of the former iteration is solved.

Acknowledgments

The author wishes to thank the Technion VLSI Lab for developing the FF grouping software, the Israel Ministry of Industry (MAG-NET program) for supporting this work, and Dr. Osnat Keren for providing the counter example. He is also grateful for the corrections and useful comments made by the reviewer.

References

- [1] E. Balas, M.W. Padberg, Set partitioning: a survey, SIAM Rev. (1976) 710-760.
- [2] L. Benini, A. Bogliolo, G. De Micheli, A survey on design techniques for systemlevel dynamic power management, IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 8 (3) (2000) 299-316
- [3] C. Chunhong, K. Changjun, S. Majid, Activity-sensitive clock tree construction for low power, in: Proc. ISLPED, 2002, pp. 279-282.
- [4] C. Chunhong, K. Changjun, M. Sarrafzadeh, Activity-sensitive clock tree construction for low power, in: Proc. ISLPED, 2002, pp. 279–282.
- [5] M. Donno, E. Macii, L. Mazzoni, Power-aware clock tree planning, in: Proc. ISPD, 2004, pp. 138–147. [6] A. Farrahi, C. Chen, A. Srivastava, G. Tellez, M. Sarrafzadeh, Activity-driven
- clock design, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 20(6)(2001) 705–714. [7] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, New York,
- 1979. [8] M.S. Hosny, W. Yuejian, Low power clocking strategies in deep submicron technologies, in: Proc. IEEE Intl. Conf. on Integrated Circuit Design and Technology, ICICDT 2008, pp. 143-146.
- [9] V. Kolmogorov, Blossom V: a new implementation of a minimum cost perfect matching algorithm, Math. Program. Comput. (2009) 43-67.
- [10] M. Muller, S. Simon, H. Gryska, A. Wortmann, S. Buch, Low power synthesizable register files for processor and IP cores, Integr. VLSI J. 39 (2006) 131-155.
- [11] V.G. Oklobdzija, Digital System Clocking-High-Performance and Low-Power Aspects, John Wiley, 2003.
- [12] G. Palumbo, F. Pappalardo, S. Sannella, Evaluation on power reduction applying gated clock approaches, in: Proc. IEEE International Symposium on Circuits and Systems, vol. 4, 2002, pp. IV-85-V-88.
- [13] W. Shen, Y. Cai, X. Hong, J. Hu, Activity and register placement aware gated clock network design, in: Proc. ISPD, 2008, pp. 182-189.
- [14] W. Shen, Y. Cai, X. Hong, J. Hu, Activity-aware registers placement for low power gated clock tree construction, in: Proc. ISVLSI, 2007, pp. 383-388.
- [15] S. Wimer, I. Koren, The optimal fan-out of clock network for power minimization by adaptive gating, IEEE Trans. on VLSI Systems 20 (10) (2012) 1772-1780.