# DESIGN AND IMPLEMENTATION OF MULTIBIT FLIPFLOP USING COMBINATIONAL LOGIC

[1]M. Ayesha, [2]K .Murali
[1]PG Scholar, [2]Head Of the Department, Department of ECE
[1,2]Narayana Engineering College, Nellore Dt , A.P.

**Abstract**— Data-driven clock gated (DDCG) and multibit flip-flops (MBFFs) are two low-power design systems that are normally treated freely. Merging these procedures into a single get-together algorithm and design stream empowers furthermore power speculation reserves. We contemplate MBFF assortment and its agreeable vitality with FF data-to-clock toggling probabilities. A probabilistic model is completed to extend the ordinary imperativeness speculation supports by get-together FFs in growing solicitation of their data-to-clock toggling probabilities. We show a front-end design stream, guided by physical configuration considerations for a 65-nm 32-bit MIPS and a 28-nm mechanical framework processor. It is seemed to achieve the power save assets of 23% and 17%, independently, stood out and designs from standard FFs. About part of the assets was a result of planning the DDCG into the MBFFs.

Index Terms— Clock gating (CG), clock network synthesis, low-power design, multibit flip-flop (MBFF).

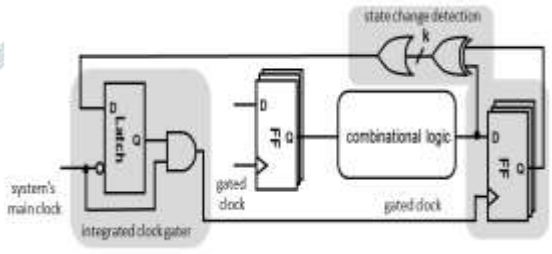
## I. INTRODUCTION

The data of digital systems are by and large secured in flip-flops (FFs), every one of which has its own specific inside clock driver. While endeavoring to diminish the clock power, a couple of FFs can be accumulated into a module called a multibit FF (MBFF) that houses the clock drivers of all the essential FFs. We mean the social affair of kFFs into a MBFF by a k-MBFF. Kapoor et al. [1] declared a 15% diminishing of the aggregate unique power in a 90-nm processor design. Electronic framework robotization instruments, for instance, Cadence Liberate, support MBFF depiction.

The upsides of MBFFs don't want free. By sharing fundamental drivers, the clock slew rate is undermined, thusly causing a larger short out stream and a more attracted out clock-to-Q propagation delay tpCQ. To cure this, the MBFF inside drivers can be braced at the cost of some extra power. It is thusly recommended to apply the MBFF at the RTL setup level to keep up a vital separation from the timing conclusion snags caused by the introduction of the MBFF at the backend design sort out. Due to the way that the ordinary data-to-clock toggling extent of FFs is pretty much nothing, which generally keeps running from 0.01 to 0.1 [2], the clock power speculation subsidizes constantly surpass the short out power discipline of the data toggling.

A MBFF gathering should be driven by authentic, helper, and FF activity considerations. While FFs gathering at the configuration level have been thought about totally, the front-end consequences of MBFF aggregate size and how it impacts clock gating (CG) has pulled in little thought. This short responds to two request. The first is the thing that the perfect piece assortment k of data-driven clock-gated (DDCG) MBFFs should be. The second is the methods by which to increase the power hold subsidizes in light of data-to-clock toggling extent (in like manner named activity and data toggling probability).

A MBFF use at the RTL justification amalgamation arrangement stage can be found in [3] for a 55-nm 230-MHz design of a system on a chip




.

**Fig. 1. DDCG integrated into a k-MBFF.**

Santos et al. [3] constrained the MBFF gathering into FFs having a place with a comparable transport. Both 2-MBFFs and 4-MBFFs were used with a 20% development in tpCQ. A dynamic power reducing of 13% was refined with some defilement in timing joining. This was helped by applying low voltage confine cells on essential courses, which to some degree extended the spillage power. The aggregate

area was extended by 2.3%, in light of the timing fixes.

To furthermore save power, [6] exhibited CG, anyway the relationship among the CG method, the FF works out, and their social occasion was not unquestionable. Wang et al. [12] depicted another postplacement algorithm that accounted absolutely to change data to evaluate the ordinary power. Regardless of the way that [6] and [12] used trading data as an assistant measure in postplacement FF gathering, our method is to use it as a basic clustering model,

Our strategy for thinking by separate is that the clock-gated MBFF should be introduced at the RTL arrangement level, in light of compositional, fundamental, and, specifically, FF activity thoughts. The essential duties of this brief are according to the accompanying:

1) a framework approach that breakers MBFF and DDCG, yielding amazing power venture stores;

2) a probability driven algorithm that restrains the typical DDCG MBFF power use.

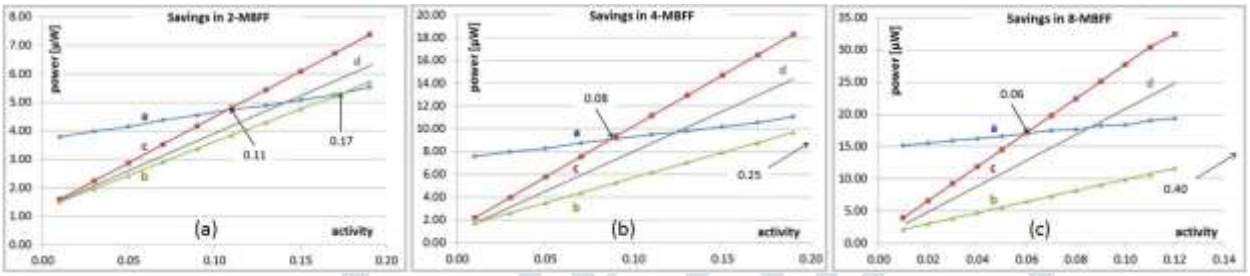



**Fig. 2. Power consumption of k 1-bit FFs compared to k-MBFF: 2-MBFF (a), 4-MBFF (b) and 8-MBFF (c). Line (a) is the power consumed by k 1-bit FFs driven independently of each other. Line (b) is the ideal case of simultaneous (identical) toggling. Line (c) is the worst case of exclusive (disjoint) toggling. Line (d) is an example of realistic toggling.(ref)**

**TABLE I**

**DEPENDENCE OF THE OPTIMAL MBFF MULTIPLICITY ON DATA-TO-CLOCK TOGGLING ACTIVITY**

| $p$ | 0.01 | 0.02 | 0.05 | 0.1 |
|---|---|---|---|---|
| $k$ | 8 | 6 | 4 | 3 |

## II. INTEGRATING CLOCK GATING INTO MBFF

Let p be the data-to-clock toggling probability. The expected energy E1 consumed by a 1-bit FF is

$$E_1(p) = \lambda_1 + \mu_1 p \ldots\ldots (1)$$

where λ1 is the energy of the FF's internal clock driver and μ1 is the energy of data toggling. In the general case of k-MBFF, let λk be the energy of the MBFF's internal clock driver and μk its per-bit data toggling energy. Assume that the FFs toggle with probability p independently of each other. It has been shown in [14] that the expected energy is

$$E_k(p) = \sum_{j=0}^{k} (\lambda_k + j\mu_k) \binom{k}{j} p^j (1-p)^{k-j} = \lambda_k + k\mu_k p$$

$$\ldots\ldots (2)$$

It is vital to take note of that toggling freedom is a cynical presumption. In all actuality, the relationship between's FF toggling yields higher vitality funds than the model in [2].

The proportion $(kE_1(p)–E_k(p))/kE_1(p)$ communicates the vitality sparing capability of k-MBFF. The coefficients λ and μ of the 65-nm 1-MBFF, 2-MBFF, and 4-MBFF were inferred with SPICE reenactments. Zero movement (p = 0) yields 35% funds for the 2-MBFF and 55% reserve funds for the 4-MBFF, while full action (p = 1.0) yields 15% investment funds for the 2-MBFF and 23% investment funds for the 4-MBFF. In ordin ary VLSI systems, the normal p does not surpass 0.1, so high reserve funds are achievable. Segment III sums up the vitality utilization display in (2) to the instance of unmistakable data toggling probabilities.

Fig. 1 illustrates a DDCG integrated into a k-MBFF. The shaded circuits reside within a library cell. Given an activity p, the group size k that maximizes the energy savings solves the equation

$$(1-p)^k \ln(1-p) C_{FF} + \frac{C_{latch}}{k^2} = 0 \ldots\ldots (3)$$

where CFF and Clatch are the clock input heaps of a FF and a latch, separately [2]. The answer for (3) for different exercises is appeared in Table I for commonplace CFF and Clatch. The above improvement does not consider the clock driver sharing, which likewise influences the ideal gathering as demonstrated as follows.

To get a handle on the power funds of a k-MBFF achievable by DDCG, Fig. 1 was recreated with SPICE for different exercises p and k = 2, 4, 8. Fig. 2(a) demonstrates the power utilization of a 2-MBFF. Line (an) is the power devoured by two 1-bit FFs driven freely of each other. The 3.8-µW power at zero movement is because of the toggling of the clock driver at each FF, and it is constantly expended paying little heed to the action.

Fig. 2(b) demonstrates the power devoured by the 4-MBFF, where line (a) relates to four 1-bit FFs driven autonomously of each other, line (b) speaks to the best instance of synchronous toggling of the four FFs, and line (c) speaks to the most pessimistic scenario of elite toggling. For zero action, the per-bit power sparing is (7.4 − 2.2)/4 = 1.3 µW, which is larger than the 1.0 µW in the 2-MBFF. Note, nonetheless, that for the most pessimistic scenario of restrictive toggling, the 4-MBFF quits sparing at 0.08 action, contrasted and 0.11 in the 2-MBFF. In the best instance of synchronous toggling, the 4-MBFF is constantly supported over the 2-MBFF. Comparable conclusions hold for the 8-MBFF appeared in Fig. 2(c). Its per-bit power putting something aside for zero movement is (15.3 − 2.5)/8 = 1.6 µW. The sparing of the 8-MBFF stops at 0.06 action in the most pessimistic scenario and at 0.40 in the best case.

## III. CAPTURING EVERYTHING IN A DESIGN FLOW

In the accompanying sections, we join the movement p and the MBFF assortment k in a plan stream went for limiting the normal squandered vitality. Fig. 2(a)– (c) shows that the power reserve funds of the 2-MBFF, 4-MBFF, and 8-MBFF, separately, are utilized. Knowing the movement p of a FF, the choice as to which MBFF measure k it best fits takes after the break lines, lines (d). To acquire the per-bit power utilization, lines (d) in Fig. 2(a)– (c), speaking to a MBFF sensible task, were separated by their particular variety. The outcome is appeared in Fig. 3.

To boost the power reserve funds, Fig. 3 isolates the scope of FF action into areas. The dark line takes after the power devoured by a 1-bit ungated FF. The triangular areas limited by the dark line and every one of the green, blue, and red per-bit lines demonstrate the sum

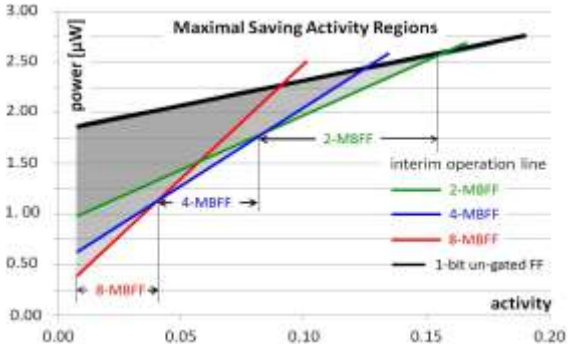



**Fig. 3. Division of the activity into ranges of maximal savings.(ref)**

of power investment funds per movement got by gathering a FF in the 2-MBFF, 4-MBFF, and 8-MBFF, individually. It demonstrates that for a very low movement, it pays to aggregate FFs into a 8-MBFF. As movement increments, there will be some point where the 4-MBFF surpasses and pays off more than the 8-MBFF. At some higher action, the 2-MBFF surpasses and pays off more than the 4-MBFF, up to a movement where the power reserve funds stops. The rest of the FFs can be assembled into ungated MBFFs, essentially to decrease the quantity of inner clock drivers. We exploit this conduct and the ideal gathering by the monotonic movement requesting appeared in Section III. The accompanying MBFF gathering algorithm is proposed.

1) Sort the n FFs such that $p1 \leq p2 \leq \cdots \leq pn$.

2) $i \leftarrow 1$.

3) Decide on optimal k by pi , based on Fig. 3.

4) If $i > n$ or $k < 2$ stop.

5) Group FFi, FFi+1,..., FFi+k−1 in a k-MBFF.

6) $i \leftarrow i + k$.

7) Go to 3.

A couple of reasonable remarks are all together. The gathering ought not cross clock areas. The clock empower signals presented by the RTL union and physically by planners are immaculate. Groupings ought to likewise consider legitimate relations and viable format concerns. One illustration is the pipeline registers of a chip, which are common possibility for MBFF usage (see Section V). It is normal that the place and course device will find bits having a place with a similar enroll near each other, though FF groups of registers having a place with particular pipeline stages will be set far from each other. FFs having a place with various pipeline registers ought to subsequently not be blended in a MBFF. Comparable contentions hold for other framework transports and

registers, for example, those putting away data, addresses, counters, statuses, and so forth. Another case is the FFs of limited state machines, whose MBFF gathering ought not cross control rationale fringes.

At last, the previously mentioned post placement MBFF clustering must think about the timing limitations, which are incorporated with their algorithms. By differentiate, the MBFF gathering algorithm does not require unequivocal timing limitations since it works at the RTL configuration level. With a specific end goal to conquer any hindrance between the RTL gathering and the gathering driven by backend timing-conclusion contemplations, we recommended proper DDCG configuration stream. The principle thought includes giving "regular" physical design orders for FF gathering by utilizing an earlier arrangement. The principle steps are depicted underneath. More points of interest can be found in [18]:

1) estimation of the FFs toggling probabilities;

2) running the situation to get preparatory favored areas of FFs in the format (dry run);

**TABLE II**

**POWER AND DELAY SAVINGS IN THE PIPELINE REGISTER OF A 32-BIT 65-nm MIPS**

| | | IF/ID | ID/EXE | EXE/MEM | MEM/WB | total |
|---|---|---|---|---|---|---|
| data-to-clock activity | sort | 0.105 | 0.0856 | 0.0711 | 0.0473 | |
| | matrix mult | 0.127 | 0.118 | 0.0799 | 0.0582 | |
| power [µW] | | 980 | 1056 | 952 | 916 | 3904 |
| savings [µW] | MBFF only | 171.0 | 240.5 | 160.7 | 129.3 | 701.5 |
| | with DDCG | 284.4 | 344.0 | 332.0 | 388.4 | 1348 |
| | post-layout | 199.5 | 273.0 | 253.3 | 232.0 | 957.8 |
| savings [%] | MBFF only | 17.4 | 22.8 | 16.9 | 14.1 | 18.0 |
| | with DDCG | 29.1 | 32.6 | 34.8 | 42.4 | 34.6 |
| | post-layout | 20.4 | 25.9 | 26.6 | 25.3 | 24.5 |

3) Using the proximity data of FFs' physical locations to constraint probability-driven grouping;

4) Adding the DDCG logic to the Verilog HDL code (done automatically by the software tool);

5) Ordinary backend flow execution.

## V. EXPERIMENTAL RESULTS

| Parameter | Existing System | Proposed System |
|---|---|---|
| Delay(ns) | 3.734 | 3.729 |

## VI. CONCLUSION

This concise proposes joining MBFFs and likelihood driven CG to expand their power reserve funds. A model using the connection between the ideal MBFF multiplicities to FF data-to-clock toggling probabilities is utilized as a part of a pragmatic plan stream, accomplishing 17% and 23% power reserve funds, contrasted and outlines with standard FFs. About portion of these reserve funds can be credited to the integration of DDCG into MBFFs.

## REFERENCES

[1] A. Kapoor et al., "Digital systems power management for high performance mixed signal platforms," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 4, pp. 961–975, Apr. 2014.

[2] S. Wimer and I. Koren, "The optimal fan-out of clock network for power minimization by adaptive gating," IEEE Trans. VLSI Syst., vol. 20, no. 10, pp. 1772–1780, Oct. 2012.

[3] C. Santos, R. Reis, G. Godoi, M. Barros, and F. Duarte, "Multi-bit flipflop usage impact on physical synthesis," in Proc. 25th IEEE Symp. Integr. Circuits Syst. Design (SBCCI), Sep. 2012, pp. 1–6.

[4] J.-T. Yan and Z.-W. Chen, "Construction of constrained multi-bit flipflops for clock power reduction," in Proc. IEEE Int. Conf. Green Circuits Syst. (ICGCS), 2010, pp. 675–678.

[5] IH-R. Jiang, C-L. Chang, and Y-M. Yang, "INTEGRA: Fast multibit flip-flop clustering for clock power saving," IEEE Trans. CAD Integr. Circuits Syst., vol. 31, no. 2, pp. 192–204, Feb. 2012.

[6] C. L. Chang and I. H. R. Jiang, "Pulsed-latch replacement using concurrent time borrowing and clock gating," IEEE Trans. Comput.- Aided Design Integr., vol. 32, no. 2, pp. 242–246, Feb. 2013.

[7] M. P.-H. Lin, C-C. Hsu, and Y-T. Chang, "Post-placement power optimization with multi-bit flip-flops," IEEE Trans. CAD Integr. Circuits Syst., vol. 30, no. 12, pp. 1870–1882, Dec. 2011.

[8] Y.-T. Shyu, J.-M. Lin, C.-P. Huang, C.-W. Lin, Y.-Z. Lin, and S.-J. Chang, "Effective and efficient approach for power reduction by using multi-bit flip-

flops," IEEE Trans. VLSI Syst., vol. 21, no. 4, pp. 624–635, Apr. 2013.

[9] S. Liu, W.-T. Lo, C.-J. Lee, and H.-M. Chen, "Agglomerative-based flip-flop merging and relocation for signal wirelength and clock tree optimization," ACM Trans. Design Autom. Electron. Syst., vol. 18, no. 3, article no. 40, Jul. 2013.

[10] M. P. H. Lin, C. C. Hsu, and Y. C. Chen, "Clock-tree aware multibit flip-flop generation during placement for power optimization," IEEE Trans. Comput.-Aided Design Integr., vol. 34, no. 2, pp. 280–292, Feb. 2015.

[11] C. Xu, P. Li, G. Luo, Y. Shi, and IH-R. Jiang, "Analytical clustering score with application to post-placement multi-bit flipflop merging," in Proc. ACM Int. Symp. Phys. Design, 2015, pp. 93–100.

[12] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo, and W.-K. Mak, "Power-driven flipflop merging and relocation," IEEE Trans. CAD Integr. Circuits Syst., vol. 31, no. 2, pp. 180–191, Feb. 2012.

[13] S-C. Lo, C-C. Hsu, and MP-H. Lin, "Power optimization for clock network with clock gate cloning and flip-flop merging," in Proc. ACM Int. Symp. Phys. Design, 2014, pp. 77–84.

[14] S. Wimer, D. Gluzer, and U. Wimer, "Using well-solvable minimum cost exact covering for VLSI clock energy minimization," Operations Res. Lett., vol. 42, no. 5, pp. 332–336, Jul. 2014.

[15] SpyGlass Power, accessed on 2016. [Online].Available:http://www.atrenta.com/solutions/spyglass-family/spyglass-power.htm

[16] Y.-T. Chang, C.-C. Hsu, M. P.-H. Lin, Y.-W. Tsai, and S.-F. Chen, "Postplacement power optimization with multi-bit flip-flops," in Proc. IEEE Int. Conf. (CAD), Nov. 2010, pp. 218–223.

[17] Hsu, Chih-Cheng, Mark Po-Hung Lin, and Yao-Tsung Chang, "Crosstalk-aware multi-bit flip-flop generation for power optimization," Integr. VLSI J. vol. 48, pp. 146–157, 2015.

[18] S. Wimer and I. Koren, "Design flow for flip-flop grouping in datadriven clock gating," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 4, pp. 771–778, Apr. 2014.