

Low power reduction techniques for Ultra Low Power Processors

Karthik Sukumar

Department of Electrical and Computer Engineering

Technical University Munich (TUM)

Munich, Germany

karthik.sukumar@tum.de

Abstract—In this report the work of J.Zhou et al. [1] and J.Tang et al. [2] on design techniques for an ultra low-power processor.

J.Zhou et al. focusses on Near-Threshold processor design for Ultra low power processors. The paper discusses the challenges, performance issue and solutions of the above.

J.Tang et al. discusses a case study by using GPS to demonstrate that an Ultra-low power processor used along with a heavy applications processor can lead to further power savings.

Index Terms—Near threshold, Ultra low power, GPS, Offload Co-processor, Interrupt service routines

I. INTRODUCTION

Power constraint portable devices (e.g. Mobile phones, wearables, IoT devices) require ultra low power processors in order to save battery life. Such low power devices enable operation with energy harvesters or small batteries. Sub and Near threshold devices have been shown to have an overall low power consumption and past works confirm that Near Threshold designs are more efficient than using clock and power gating.

Near threshold designs refer to operating the transistor slightly above the threshold region.

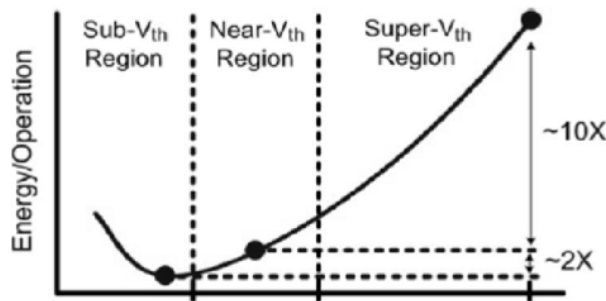


Fig. 1. Sub/Near-threshold operation. Source: [1]

The major advantage of using the near threshold devices is the decrease in dynamic power per operation. as it can be seen from Fig. 1

Clock tree networks consume a large part of the overall power needed by an IC. The named values vary from 25% [4] to 50% [5]. There are several possibilities to reduce the power consumed by the clock network.

One of the most popular methods is voltage scaling due to its quadratic savings ($P_{dyn} = fC_L V_{DD}^2$). The used voltage in ultra-low voltage (ULV) applications is close to the threshold voltage (V_T). This saves energy, but makes the design less robust and susceptible to MOSFET process variations [6]. The way to achieve a low-voltage clock tree with a minimized and well-defined skew and with a well-controlled slew is discussed in [1]. The findings of [1] are presented in section II.

Another approach to reduce the power consumed by the clock network is to optimize the network topology. While the skew control was well studied in two dimensional ICs [7], the optimal topology in nowadays 3D stacked ICs is still being investigated. Through-silicon vias (TSVs) are used for the connection of clock networks on different dies. The impact of the TSV resistance-capacitance (RC) and the TSV count on the clock power, clock skew and clock skew is subject of [2]. The results of [2] are presented in section III.

II. ULTRA-LOW VOLTAGE CLOCK NETWORK DESIGN

Generally, there are multiple ways to design a clock network. These ways are shown in Fig. 2. Connecting all sinks (a sink can be anything that needs a clock signal, e.g. flip-flops) without attempting to balance the different paths is called "signal-route". In contrast to that, the H-trees try to take this into account. It is obvious that the path length from the clock source to each sink becomes more equal the higher the H-tree's level is.

Furthermore, there is a difference between unbuffered and buffered H-trees. In unbuffered H-trees, the whole circuit is driven by a single driver, whereas in a buffered H-tree several smaller buffers are inserted to drive the tree. Normally, designers tend to use the buffered version, because then each driver has to drive a smaller RC-value and therefore the interconnect delay is mitigated, but as we will see later on this is not longer true for ULV applications.

A. Clock networks behaviour without considering MOSFET process variations

Energy consumption and clock skew can be simulated using SPICE. The simulation results for a $0.18\mu m$ CMOS framework with $0.3V$ supply voltage are shown in Fig. 3. The skew decreases with the levels in tree as the paths to the sinks become more equal. This holds for both, the buffered and

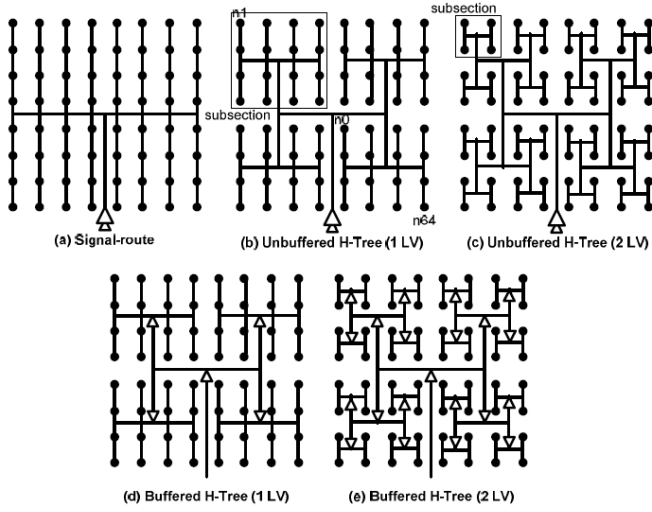


Fig. 2. Network topologies. Source: [1]

unbuffered trees. The energy increases with the levels due to the overall wire length becoming larger. For higher tree levels, the buffered version consumes less energy because the smaller buffers are more energy efficient in driving longer wires.

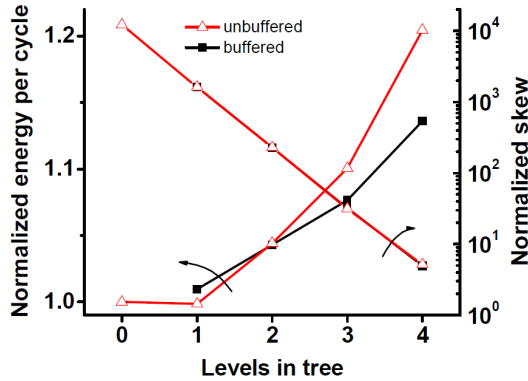


Fig. 3. Energy and skew over levels in tree. Simulated using SPICE without considering MOSFET process variations. Source: [1]

B. Clock networks behaviour with considering MOSFET process variations

Since MOSFET process variations (e.g. random V_{th} mismatch) have an exponential effect on gate delay at ULV circuits [?] this has to be considered. For this reason M. Seok et al. [1] performed Monte Carlo simulations with random MOSFET mismatch. The results are shown in Fig. 4. In contrast to the simulations without variations, the buffered trees perform much worse (Fig. 4 (a)) in skew because the buffer delay is not anymore cancelled among buffers and starts to contribute to skew. The skew variability (σ/μ) first decreases as more buffer stages are used with increasing tree level and thus the variations are averaged. But after level three it starts to increase again as smaller buffers are more sensitive to process variations.

The skew behaves similarly. The unbuffered version shows a good robustness while the buffered version behaves worse. The skew variability increases with the levels in tree. There is no averaging effect for the skew variability because the skew is mainly determined by the last buffer stage before the sink.

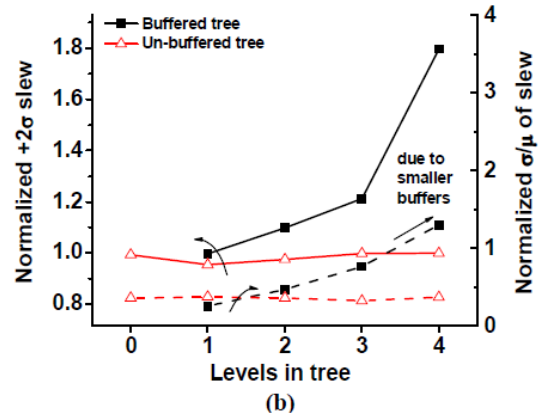
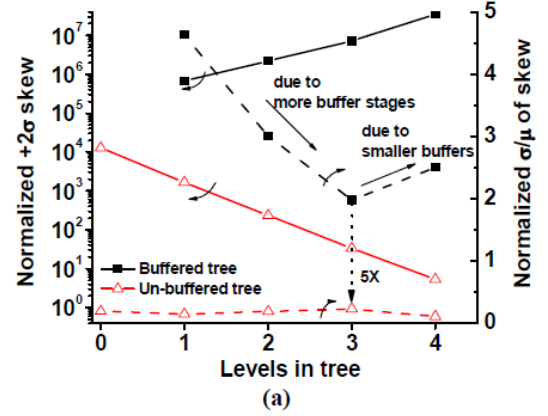


Fig. 4. Skew and slew over levels in tree with considering MOSFET process variations. Source: [1]

C. Adding buffers in ULV regimes

In higher voltage regimes adding buffers to long interconnects often improves the interconnection delay (see Fig. 5). For ULV applications this is not longer true, because the delay penalty for adding buffers is larger than the reduction of wire RC. As shown in Fig. 5 the version without repeater still performs better for a 30mm wire.

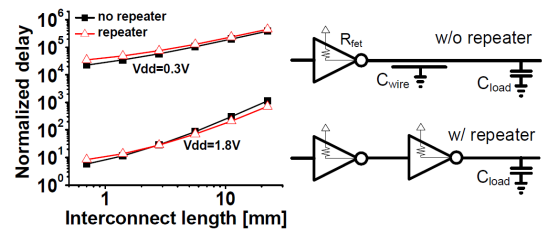


Fig. 5. Driving long interconnects with and without repeaters at 0.3V and 1.8V. Source: [1]

III. CLOCK NETWORK DESIGN FOR THROUGH-SILICON VIA (TSV) BASED 3D ICs

Today's ICs are often built as 3D stacked ICs. The different dies are connected using through-silicon vias (TSVs). The 3D clock tree synthesis problem can thereby be formulated as follows: Given are a set of sinks, a TSV bound (the user defined maximum of TSVs between adjacent dies), a clock source location and the wire and TSV parasitics. Now a clock network has to be constructed such that all sinks are connected to the clock, the TSV count is under the TSV bound, the skew is minimized, the slew is below the constraint and the wire length and the clock power are minimized.

To solve this problem X. Zhao et al. [2] developed a 3D clock tree synthesis algorithm that consist of two steps:

- 1) 3D abstract tree generation to determin how to connect the sinks to each other.
- 2) Slew-aware buffering and embedding to determine the exact routing topology.

A. 3D Abstract Tree Generation

The abstract tree generation is an iterative algorithm. In each step, the set of sinks is divided into two subsets using two different methods:

- Z-cut: if the TSV bound is one, the set of sinks is divided into two subsets such that all sinks from the same die belong to the same subset. The two new subsets are connected using a TSV. When the sinks are located on N dies, $N - 1$ iterations of Z-cuts are needed.
- X/Y-cut: if the TSV bound is larger than one or all sinks in the set belong to the same die. The set is divided geometrically by a horizontal line. The Z-dimension is ignored. The TSV bound before the X/Y-cut is then redistributed to the new subsets.

An example for a X/Y-cut followed by two Z-cuts is shown in Fig. 6.

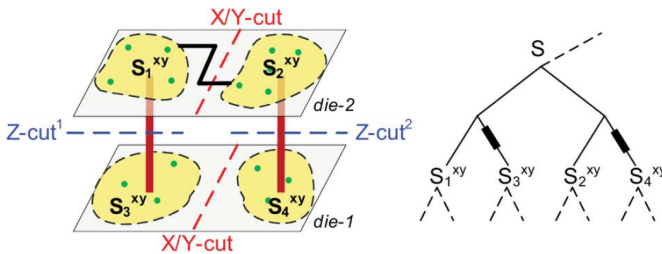


Fig. 6. X/Y-cut followed by two Z-cuts. Source: [2]

The algorithm is finished when each sink is in its own set. The result of the algorithm for a two die IC under various TSV bounds is shown in Fig. 7.

After the binary tree is calculated, the slew-aware buffering and embedding takes place.

B. Impact of the TSV count on the clock power

The more TSVs are used, the smaller becomes the overall wire length. This leads to a smaller wire RC and therefore

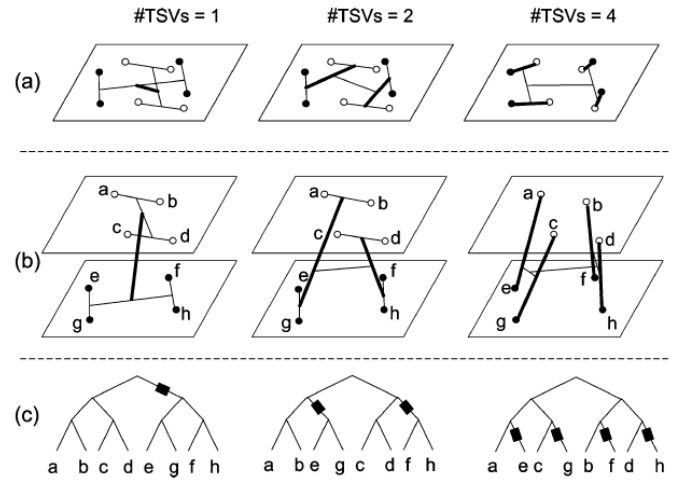


Fig. 7. The 3D abstract tree generation under various TSV bounds. (a) shows a top view, (b) shows a 3D view and (c) show the resulting binary tree where the squares denote TSVs. Source: [2]

the circuit consumes less power. But after a certain point the capacitance of the inserted TSVs are too large what then results in an increasing power consumption. Hence, there is a minimum power point for a certain number of TSVs. This point can be found by sweeping the TSV bound in the simulation as it was done for Fig. 8.

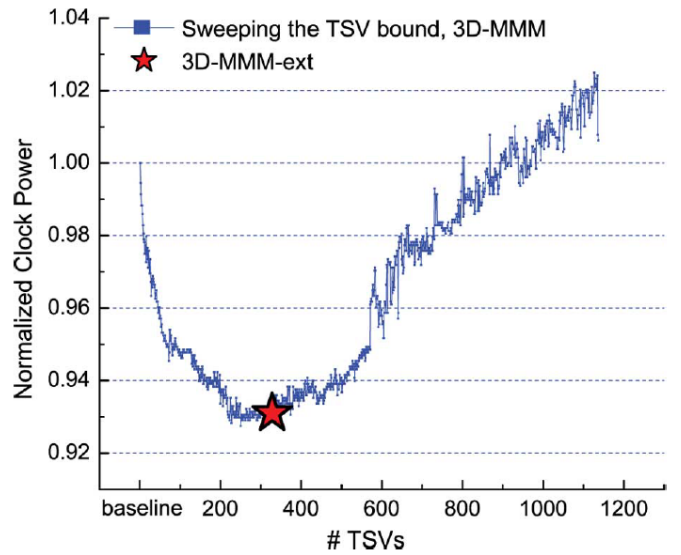


Fig. 8. Clock power over TSV count for a two die stack using 100fF TSVs. The red star denotes the predicted optimum power point found by 3D-MMM-ext algorithm which was developed in [2]. Source: [2]

The exhaustive search by sweeping the TSV bound is often not practicable due to its time-consuming runtime. For this reason X. Zhao et al. [2] developed the 3D-MMM-ext algorithm that can predict the minimum power point in a good manner. Moreover, X. Zhao et al. showed that using the optimal TSV count can save up to 36% of the total power consumed by six-die stacked IC.

IV. CONCLUSION

As a conclusion one can say that a well designed clock network can save a lot of power. This can be achieved by voltage scaling, but in this case the clock network has to be designed in different way than it would be done for a super-threshold regime. M. Seok et al. showed that for ULV applications unbuffered clock trees consume less power and are in addition to that more robust to MOSFET process variations and therefore show a better skew and slew behaviour.

For nowadays 3D staced ICs, there is an optimal number of TSVs regarding the power consumption. This is because more TSVs can reduce the overall wire lenght (and therefore the wire RC), but too many TSVs lead to a too large capacitance.

So we can finally conclude that the consumed power by the clock can drastically be reduced by choosing the optimal routing topology. Afterwards, the power can still be reduced by measures such as voltage scaling.

REFERENCES

- [1] J.Zhou, T.H.Kim and Y.Lian, "Near-threshold processor design techniques for power-constrained computing devices," 2017 IEEE 12th International Conference on ASIC (ASICON), Guiyang, China, 2017, pp.920-923
- [2] J. Tang, C. Liu, Y. L. Chou and S. Liu, "OCP: Offload Co-Processor for energy efficiency in embedded mobile systems," 2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors, Washington, DC, 2013, pp. 107-110
- [3]
- [4]
- [5]
- [6]
- [7]