Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

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

Thermal aware design and comparative analysis of a high performance 64-bit adder in FD-SOI and bulk CMOS technologies

Can Baltacı[*], Yusuf Leblebici

LSM, STI, EPFL, 1015 Lausanne, Switzerland

ARTICLE INFO

Keywords:
Self-heating
Thermal modelling
Thermal simulation
Bulk vs FDSOI
64-bit adder in dynamic logic

ABSTRACT

Thermal behaviours of high-performance digital circuits in bulk CMOS and FDSOI technologies are compared on a 64-bit Kogge-Stone adder designed in 40 nm node. Temperature profiles of the adder in bulk and FDSOI are extracted with thermal simulations and hotspot locations are studied. The influence of local power density on peak temperature is examined. It is shown that high power density devices have significant influence on peak temperature in FDSOI. It is found that some group of devices that perform the same function are the most prominent heat generators. A modification on the design of these devices is proposed which decreases the hotspot temperatures significantly.

1. Introduction

The demand for increasing the performance of high speed digital circuits brings the need for smaller and faster implementations [1,2]. However, as the clock frequencies increase owing to smaller technology nodes, the circuits consume power in higher densities. Consequently, the temperature levels are elevated and the thermal issues become the bottleneck of the circuits by altering the performance and decreasing the lifetime. On the performance side, the temperature induced reduced mobility decreases the device current and the maximum speed of operation [3]. Moreover, the threshold voltage decreases with the temperature and this results in higher leakage current and power consumption [4,5]. Higher power consumption brings higher temperature and this might result in thermal runaway where the die fails due to the uncontrolled increase in the temperature. Although thermal runaway does not happen, the chip might settle down to a higher temperature, which would degrade the performance as well as the reliability of the chip [6]. Electromigration phenomena is another reliability problem related to temperature where the metal interconnects are broken due to diffusion or flow of atoms under very high current densities at high temperatures [7,8]. All of the mentioned problems show that having a reliable and high performance chip is not possible without considering the thermal behaviour of the design. This brings another aspect into the design space, which is the self-heating.

Self-heating became a more critical problem especially after the introduction of the modern MOSFET device geometries like FinFET and Fully Depleted Silicon on Insulator (FDSOI) [9]. Previously, it was reported that the peak temperature of the FDSOI devices is located close to the drain end of the device [1,10,11] and the peak temperature value in FDSOI FETs is found to be much higher than the one in the conventional bulk MOSFETs [12,13]. The higher peak temperature of the FDSOI structure is mainly due to the thermal behaviour of its constitutive materials. The thermal conductivity of the SiO_2 isolation layer is two orders of magnitude lower than the thermal conductivity of the bulk Si. Moreover, the thermal conductivity of Si thin film, where the devices are generating heat, is one order of magnitude less than the thermal conductivity of bulk Si [1]. Additionally, the boundary between Si and SiO_2 creates a finite interface thermal resistance, [14,15] which is equal to the thermal resistance of a SiO_2 layer with a thickness of 20 nm [16]. Due to the mentioned facts, the dissipated power in FDSOI devices does not find a high conductance diffusion path. Consequently, the generated heat turns into temperature in nanometer scale local spots which are comparable to the dimensions of transistors in FDSOI. However, not all the devices settle down to very high temperature values in an implementation. The devices which consume the highest amount of power per unit area are the hottest ones especially in FDSOI. As a result of this, a design which contains devices with large differences in their power densities create very prominent temperature hotspots and large temperature gradients. By performing a detailed power density analysis, the critical ones can be eliminated from the others; and by performing some modifications on their design, the peak temperature and the high temperature gradients can be reduced. Recently, during the implementation of a 5 GHz processor, high switching factor nets were identified during functional simulation to avoid micro hotspots at the individual gate level, caused by device self-heating [17]. As a solution, the maximum output load capacitances of

[*] Corresponding author.
E-mail addresses: can.baltaci@epfl.ch (C. Baltacı), yusuf.leblebici@epfl.ch (Y. Leblebici).

http://dx.doi.org/10.1016/j.vlsi.2017.03.001

the gates driving these nets are reduced and these gates are placed away from the other gates driving such nets in order to avoid excessive heating and have uniform temperature distribution overall the circuit.

In this work, we intend to emphasize the correlation of the nanometer scale hotspots and the power density of individual devices by observing the temperature profile of high performance circuits implemented in bulk and FDSOI. For this purpose, a 64-bit parallel prefix adder is designed and implemented in a commercially available 40 nm CMOS bulk technology. The power dissipation of each device in the circuit is observed under randomly applied input vectors. The resulting power dissipation output is provided as an input to thermal simulations for observing the temperature profile of the overall block. HotSpot tool [18] is used for modelling the thermal behaviours of bulk and FDSOI geometries. The devices situated on the highest temperature locations are found and examined. It is observed that self-heating in FDSOI is much more prominent when compared to bulk since the local hotspots have sizes comparable to the size of the devices and the generated heat is directly converted to temperature in FDSOI. Consequently, the highest temperature values occur on the devices which have the highest power density. Finally, a solution for decreasing the temperature of the hotspots is proposed. It is shown that the peak temperature of the design in FDSOI can be decreased significantly with a cost of an insignificant increase in the area and parasitic capacitances.

In Section 2, the performance parameters of the implemented 64-bit parallel prefix adder are given and its architecture is explained in detail. In Section 3, bulk and FDSOI thermal simulation results and the temperature profiles of the designed 64-bit adder is provided. In Section 4, the correlations between the devices with the peak temperature and their functions are shown. Finally, in Section 5, the summary of the work and the conclusions are provided.

2. Implemented block

The parallel prefix adder is implemented with Kogge-Stone technique [19] where radix-4 and sparsity-4 options are used [20,21]. The entire 64-bit Kogge-Stone adder block is designed with full custom design approach (Fig. 1). The block is primarily optimized to obtain the lowest possible critical path delay while having the lowest possible power consumption and area. Finally, a delay (clock to sum) of 148 ps is obtained under 900 mV power supply voltage. The block contains 10922 nMOS and pMOS devices and the resulting area of the block is around 2200 μm^2. The average power dissipation of the block is 12 mW and the average power density is 548 W/cm^2 under a clock frequency of 2.5 GHz with 50% duty cycle. This corresponds to 200 ps evaluation time which is 52 ps more than the critical path delay. The detailed block diagram of the implemented 64-bit Kogge-Stone Parallel Prefix Adder can be seen on Fig. 2. The interconnect lines, signal names and the blocks on the critical path are indicated by red colour. The block consists of three main building blocks which are Propagate-Generate Signal Generator, Propagate-Generate Signal Merge and *4-bit* Carry Select Adders (CSA). The detailed explanation of the architecture of these blocks are given in the following sub-sections.

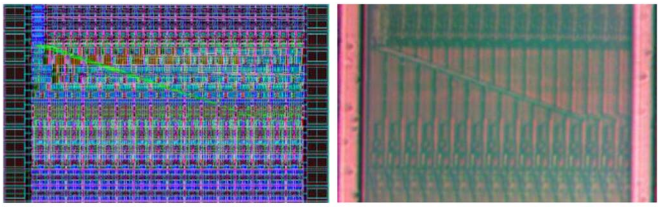
(a) Layout (b) Micrograph

Fig. 1. Full custom layout and die micrograph of the 64-bit adder block (54.8 μm×40 μm) implemented in 40 nm technology. (a) Layout (b) Micrograph.

2.1. Propagate-generate signal generator

In parallel prefix adders, the reduction in the delay time is provided by merging *Propagate* (*P*) and *Generate* (*G*) signals in a parallel fashion to obtain the values of the *Carry Out* signals. For that reason, *Propagate* and *Generate* signals are generated before the merging step according the Boolean expressions given by

$$\overline{P_i} = \overline{A_i + B_i} \tag{1}$$

$$\overline{G_i} = \overline{A_i \bullet B_i} \tag{2}$$

where the subscript *i* is the bit number of the input values *A* and *B*. In this implementation, instead of *P* and *G* signals, and signals are generated in the *Propagate-Generate Signal Generator* block, mainly for decreasing the logic depth and increasing the clock frequency. The logic gates are implemented with N-Domino Logic with the clocked footing devices [22].

2.2. Propagate-generate signal merge

The *Propagate-Generate Signal Merge* block is the heart of the overall 64-bit Adder implementation since the evaluation time of this part has an important influence on the speed of the overall block. In this block, *P* and *G* signals are merged to get the *Carry Out* information of different stages in the addition. On Fig. 2, the blue coloured circles indicate a logic gate which performs the merging operation of four *P* and four *G* signals, where the radix option is set to 4 for further decreasing the critical path delay by decreasing the logic depth [20,23]. The Boolean expression of these functions are shown by (3) and (4) where the subscript *i*:*i*−3 indicates that the output signals are the merged *P* and *G* signals from the bits *i* to *i*−3.

$$P_{i:i-3} = P_i \bullet P_{i-1} \bullet P_{i-2} \bullet P_{i-3} \tag{3}$$

$$G_{i:i-3} = (G_i + G_{i-1} \bullet P_i) + (G_{i-2} + G_{i-3} \bullet P_{i-2}) \bullet P_i \bullet P_{i-1} \tag{4}$$

The radix-4 option provides the advantage of merging four signals with a single logic gate. However, (3) and (4) shows that the implemented CMOS logic gate will be quite complex and it will contain 4 transistors in series. This fact will in turn decrease the speed of the logic gate especially for the advanced technology nodes where the power supply voltages are equal to or below 1 V. Another possibility to implement a radix-4 PG-Merge gate is to cascade two radix-2 PG-Merge gates. Hence, (3) and (4) can be written as

$$P_{i:i-3} = (P_i \bullet P_{i-1}) \bullet (P_{i-2} \bullet P_{i-3}) = P_{i:i-1} \bullet P_{i-2:i-3} \tag{5}$$

$$G_{i:i-3} = [G_i + G_{i-1} \bullet P_i] + [(G_{i-2} + G_{i-3} \bullet P_{i-2}) \bullet (P_i \bullet P_{i-1})] = G_{i:i-1} + G_{i-2:i-3} \bullet P_{i:i-1} \tag{6}$$

where the cascading of two radix-2 PG-Merge gates can be explicitly seen. The cascading approach has the disadvantage of having a logic depth of two when compared to the approach shown by (3) and (4); however, the series resistance in each gate is quite relaxed. At this point, the delay performance can be questioned since both approaches have their own advantages and disadvantages. To observe the faster solution, the same 64-bit Adder is implemented with the both approaches. It is observed that the cascading approach is unequivocally faster than the single gate approach.

As indicated in Section 2.1, only and signals are available at the inputs of the *Propagate-Generate Signal Merge* block. Consequently, (7) and (8) are used for implementing the radix-4 PG-Merge gates where both the inputs and the outputs are negated.

$$\overline{P_{i:i-3}} = \overline{\overline{(\overline{P_i} + \overline{P_{i-1}})} \bullet \overline{(\overline{P_{i-2}} + \overline{P_{i-3}})}} = \overline{P_{i:i-1} \bullet P_{i-2:i-3}} \tag{7}$$

$$\begin{aligned}\overline{G_{i:i-3}} &= \{\overline{\overline{G_i} \bullet (\overline{G_{i-1}} + \overline{P_i})} + [\overline{\overline{G_{i-2}} \bullet (\overline{G_{i-3}} + \overline{P_{i-2}})} \bullet \overline{(\overline{P_i} + \overline{P_{i-1}})}]\}' \\ &= \overline{G_{i:i-1} + G_{i-2:i-3} \bullet P_{i:i-1}}\end{aligned} \tag{8}$$

All radix-4 PG Merge gates in the *Propagate-Generate Signal*