Power Optimization of FPGA Interconnect Via Circuit and CAD Techniques

Safeen Huda and Jason H. Anderson
Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, ON Canada
{safeen|janders}@ece.toronto.edu

ABSTRACT

We target power dissipation in field-programmable gate array (FPGA) interconnect and present three approaches that leverage a unique property of FPGAs, namely, the presence of *unused* routing conductors. A first technique attacks dynamic power by placing unused conductors, adjacent to used conductors, into a high-impedance state, reducing the effective capacitance seen by used conductors. A second technique, charge recycling, re-purposes unused conductors as charge reservoirs to reduce the supply current drawn for a positive transition on a used conductor. A third approach reduces leakage current in interconnect buffers by pulsed-signalling, allowing a driving buffer to be placed into a high-impedance stage after a logic transition. All three techniques require CAD support in the routing stage to encourage specific positionings of unused conductors relative to used conductors.

CCS Concepts

•**Hardware → Reconfigurable logic and FPGAs; Programmable interconnect;**

Keywords

FPGA, physical design, low-power, routing

1. INTRODUCTION

Field-programmable gate arrays (FPGAs), long used in communications and industrial applications, are gaining traction in the high-performance computing space, where they are used alongside traditional processors to implement accelerators that provide improved computational throughput and energy efficiency. The recent $16B acquisition of Altera (a leading FPGA vendor) by Intel, and the use of FPGAs by Microsoft for Bing search acceleration [10] assure the presence of FPGAs in future datacenters. At the other end of the computing spectrum, however, in low-power/mobile/IoT, FPGAs have seen far less success, primarily because of their high power consumption, which recent work [6] has shown to be multiple times higher than fixed-function ASICs. In this paper, we present combined architecture, circuit, and CAD techniques for reducing power consumption in FPGA interconnect, which accounts for the majority of dynamic and leakage power in FPGAs [13, 12].

We present three techniques for reducing FPGA interconnect power consumption, that hinge on a unique property of FPGAs – the property that a given application circuit implemented in an FPGA uses only a fraction of the FPGA's underlying interconnect resources, leaving many resources unused. The presented techniques leverage the unused resources in several different ways. Two of the techniques presented target dynamic power; the third targets leakage power. All three techniques are tied together in that they require very similiar CAD support in the routing stage of the flow. Specifically, the three techniques require that, to the extent possible, used conductors in the FPGA interconnect are physically adjacent to unused conductors.

The dynamic power consumed by a logic signal in a CMOS circuit is given by: $\alpha \cdot C \cdot V_{DD} \cdot V_{swing}$, where α represents the signal's toggle frequency, C is capacitance, V_{DD} and V_{swing} the supply and swing voltage, respectively. In a first approach [4], we attack the C term by reducing the intra-layer coupling capacitance a used conductor "sees" towards an unused neighbour. This is achieved by a lightweight switch buffer design that permits the unused neighbour to be placed into a high-impedance state, resulting in the coupling capacitance being *in series* with other capacitances, thereby lowering effective capacitance. In a second approach, we target V_{swing} by *charge recycling* to/from neighboring unused routing conductors [3]. The key idea is to pair together routing conductors in the FPGA routing architecture such that, when one of the pair is unused, it can serve as a charge reservoir for the used conductor. A portion of the charge normally dissipated to ground on a falling transition is instead stored in the reservoir, to be recycled to the used conductor on a rising transition, reducing V_{swing} and saving power.

We then present a new approach for reducing leakage power in the FPGA interconnect structures. We propose pulsed-signalling, wherein a driving interconnect buffer "pulses" its intended signal value to a receiving buffer for a short time interval, after which it enters a high-impedance (low-leakage state), instead of maintaining the value as in traditional CMOS signalling. To the extent possible, conductors adjacent to used conductors must be kept unused, to mitigate noise and ensure the logic value "captured" by the receiving buffer is not disrupted. Again, this approach relies on noise-reduction support in the routing stage of the FPGA CAD flow. To the authors' knowledge, this is the first work to propose pulsed-signalling in the FPGA context.

The remainder of this paper is organized as follows: Section 2 provides background material on FPGA interconnect architecture. Section 3 summarizes the charge recycling and capacitance reduction work for dynamic power reduction. Section 4 introduces the pulsed-signaling for leakage reduc-

ISPD'16, April 03-06, 2016, Santa Rosa, CA, USA
© 2016 ACM. ISBN 978-1-4503-4039-7/16/04...$15.00
DOI: http://dx.doi.org/10.1145/2872334.2872341

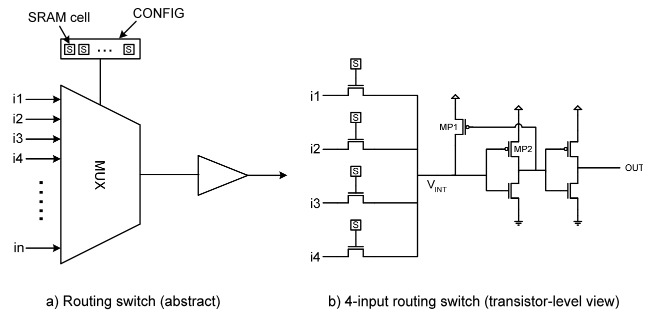
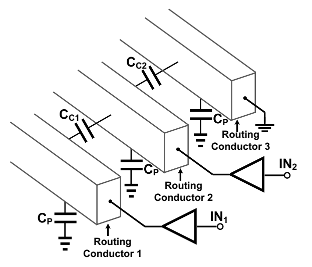
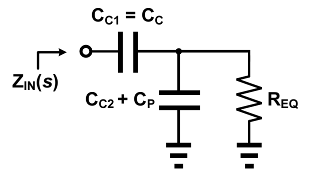

SRAM cell
CONFIG
S S ⋯ S
i1
i2
i3
i4
in
MUX
a) Routing switch (abstract)
S
i1
i2
i3
i4
MP1
MP2
V_INT
OUT
b) 4-input routing switch (transistor-level view)

Figure 1: Interconnect switch.



(a) Routing buffers and their respective loads.



(b) Simplified model of the impedance seen by a routing buffer looking towards an adjacent conductor

Figure 2: Interconnect impedance modeling

tion. CAD support is discussed in Section 5. Experimental results appear in Section 6, followed by conclusions and future work.

2. BACKGROUND

FPGA interconnect consists of metal wire segments that may be programmably connected to logic block pins and to one another through interconnect switches. Fig. 1(a) shows an FPGA routing switch comprised on a multiplexer (MUX), buffer and SRAM configuration cells. The data inputs to the MUX attach to logic block output pins and/or other routing conductors. The SRAM cell contents select a particular input to drive to the switch output, which attaches to a logic block input or a routing conductor. The transistor-level view in Fig. 1(b) illustrates that the MUX is typically built with NMOS transistors and the buffer is "level-restoring" via transistor $MP1$, which serves to pull the buffer input to the supply rail when a logic-1 is passed through the NMOS-based MUX. State-of-the-art Xilinx and Altera FPGAs use such unidirectional buffered switches.

The two main commercial FPGA vendors use a variant of the PathFinder negotiated congestion routing algorithm [9]. In PathFinder, the device interconnect is represented as a directed graph, $G(V, E)$, where each $v \in V$ represents a conductor and each $e \in E$ represents a programmable connection between two conductors. Routing a signal from its driver to its loads is therefore translated into a minimum-cost search in G between the vertices corresponding to the driver/load pins. The PathFinder algorithm allows shorts between signals at intermediate stages of the routing process. Congestion removal is handled via cost adjustments based on node oversubscription, along with rip-up and re-route. Signal timing criticality information is used to encourage the routing of a design's timing-critical connections on low-delay paths through the FPGA fabric.

3. DYNAMIC POWER REDUCTION TECHNIQUES

3.1 Reducing Effective Interconnect Capacitance

We recently proposed a technique to reduce the effective capacitance loading a routing wire [4]. The approach is inspired by an analysis of the capacitance seen at the output of a routing buffer: this capacitance is mainly the parasitic capacitance of the driven metal interconnect wire. The different components of the total parasitic capacitance for a routing conductor are annotated in Fig. 2a.

The figure shows that the capacitance of a routing conductor is due to the coupling capacitance (C_C) between adjacent metal conductors on the same metal layer, and the plate capacitance (C_P), due to overlap with other conductors on adjacent layers (the adjacent layer is depicted as a ground plane). Currently, C_C dominates the total interconnect capacitance: ITRS projections indicate C_C is approximately twice as large as C_P (and this ratio is projected to continue to increase) [5]. This fact is leveraged to reduce the effective capacitance of an interconnect conductor, since the total capacitance is equal to $C_{tot} \approx 2C_C + 2C_P$, if C_C is reduced it will result in a significant reduction in C_{tot}. Our technique to reduce C_C is based on an analysis of the capacitance a routing conductor "sees" when looking towards an adjacent routing conductor on the same metal layer.

Fig. 2a shows three routing conductors, with their respective driver circuits. Routing conductor 1 is coupled with conductor 2 through coupling capacitor C_{C1} and similarly, conductor 2 is coupled with conductor 3 through C_{C2}. From the perspective of conductor 1, conductor 2 and coupling capacitors C_{C1} and C_{C2} together form the equivalent, approximated circuit shown in Fig. 2b. In the figure, R_{eq} is the effective output resistance of the driver driving conductor 2, and it is assumed that the effective driver resistance for the buffer driving conductor 3 is near 0 (the conductor is grounded). The input impedance of this circuit has the following transfer function in the Laplace (s) domain:

$$Z_{in}(s) = \frac{R_{eq}(2 \cdot C_C + C_P)s + 1}{C_C s[R_{eq}(C_c + C_p)s + 1]} \tag{1}$$

where it is assumed that $C_{C1} = C_{C2} = C_C$. We now note that if $R_{eq} >> 1/(C_C + C_P)$, $Z_{in}(s) \approx (2 \cdot C_C + C_P)/(C_C(C_C + C_P))$. This implies that if conductor 2 is unused, and its driver impedance can be set large, the effective impedance seen from conductor 1 looking towards conductor 2 is that of capacitor C_{C1} in *series* with the parallel combination of capacitors C_{C2} and C_P, which means that the effective parasitic loading on routing conductor 1 can therefore be *reduced.* A buffer with a reasonably small effective output resistance can be made to have a much larger output resistance if the buffer can be put into a tristate mode. Putting a buffer into tristate mode requires disconnecting all paths to either V_{DD} or GND, which ensures that there are no low-impedance paths to either rail.

In [4], we proposed to reduce power dissipated in active routing conductors by placing their adjacent unused conductors into tristate mode. Parasitic capacitance reduction is observed only when an active routing conductor is adjacent to one or more unused routing conductors. Since the proposed technique requires routing buffers to be tristateable, we also propose a tristate buffer topology with minimal area-overhead, specially tailored for FPGA interconnect drivers.

3.2 Charge Recycling

Another technique we proposed that exploits unused routing conductors is charge recycling [3]. The idea is to make use of the energy dissipated in previous transitions, and thus to actively recycle "used" charge. In conventional logic circuits, when a conductor makes a falling transition, the stored energy on the wire – equal to $CV_{DD}^2/2$ where C is the wire capacitance – is completely dissipated. In contrast, charge recycling stores some of that (normally) dissipated charge in secondary nodes (capacitors). This charge can then be delivered to a wire which is making a rising transition, reusing some of the energy that would have been otherwise lost. The overall principle behind the technique is shown in Figs. 3 and 4.

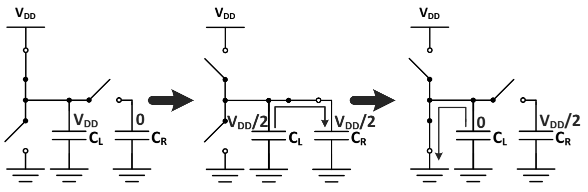


Figure 3: Charge recovery from a "1" to "0" transition

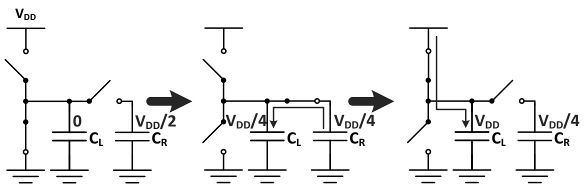


Figure 4: Charge recycling during a "0" to "1" transition

The figures show two capacitors, C_L, the load capacitor, and C_R, a reservoir capacitor used to store recovered charge. Switches connect C_L to either V_{DD} or GND, and a third switch can be used to connect the two capacitors together. In Fig. 3, we see that C_L is initially connected to V_{DD}, and there is no charge stored in C_R. During a falling transition at C_L, initially the circuit undergoes a *charge recovery phase*, where C_R and C_L are connected (both V_{DD} and GND are disconnected from C_R). Given that the two capacitors have equal capacitance, through charge sharing, half of the charge initially stored in C_L will be transferred to C_R, and thus the voltages of these capacitors will settle to $V_{DD}/2$. After the charge recovery phase, the two capacitors are disconnected, and C_L is connected to GND, and thus fully discharged.

In Fig. 4, C_L is initially connected to GND, and the voltage at C_R is equal to $V_{DD}/2$ as a consequence of the charge recovery phase of a preceding falling logic transition. In a rising transition, the circuit initially undergoes a *charge recycling phase* where C_R and C_L are connected to one another while both the V_{DD} and GND rails are disconnected. Through charge sharing, the voltage at C_L will rise (while the voltage at C_R will fall) to $V_{DD}/4$. After the voltage at C_L has settled, C_L and C_R are disconnected from one another, and the C_L is connected to V_{DD} to complete the full transition to logic-1. Note that in this example, the amount of charge actually drawn from the V_{DD} rail is equal to $0.75 \cdot C_L \cdot V_{DD}$ as opposed to $C_L \cdot V_{DD}$. That is, there is an immediate 25% reduction in energy.

To reduce power consumption in the routing network, we propose an FPGA architecture wherein unused routing conductors are re-purposed as charge reservoirs, where they store charge from falling transistors on the used routing conductors to which they are paired , and permit that charge to be recycled on rising transitions.

To implement the desired behavior, we designed the buffer circuit shown in Fig. 5, which includes an intermediate stage to apply the appropriate signals to the gates of $M1$ and $M2$ (which comprise the buffer's output stage), based on the operating mode of the buffer. For a full description of the circuit structures, the reader is referred to [3]. When operating in charge recycling mode, the technique allows dynamic power of the buffer to be reduced by ~26% which compares favourably to a theoretical maximum power reduction of 33%.

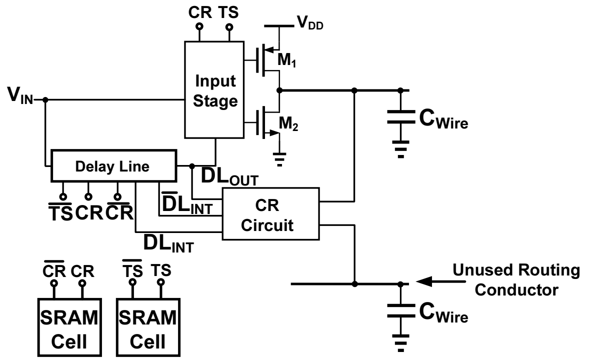


Figure 5: Charge recycling routing buffer.

4. PULSED-SIGNALLING FOR LEAKAGE POWER REDUCTION

Routing conductors in FPGAs are connected to a significant number of routing multiplexers which typically have a large number of inputs to ensure sufficient routing flexibility. For the sake of optimizing area-delay tradeoff, routing multiplexers are usually built with pass transistors in a two-stage topology [1]. However, these routing multiplexers contain multiple leakage paths, resulting in considerable leakage power being dissipated in FPGA interconnect. In the face of impending issues in the sub-10nm era, such as direct source-drain tunneling (DSDT), it has become critical to find means to reduce the leakage dissipated in FPGA interconenct without sacrificing routing flexibility.

To reduce leakage, we begin by observing that the output impedance of the routing buffers in the interconnect network plays a crucial role in the leakage currents which flow from/to routing MUXes. Conventional routing buffers provide low impedance paths (either from V_{DD} or to GND), and it is through these paths that leakage currents flow. If instead we are able to increase the impedance of these paths, leakage current may be reduced. Simply increasing the output impedance of routing buffers is not an acceptable solution, as this would lead to performance penalties. Note however that a buffer's output impedance only has to be low when the buffer is in the process of transmitting data (and its output is transitioning); if at all other times, the output impedance is kept high, leakage power will be reduced. To achieve this desired behaviour, we propose to replace conventional routing buffers in FPGAs (see Fig. 1)) with routing buffers which are tristated when they are not in the process of sending data, and dynamically activated following a transition at their inputs (when they need to send data). In the following sections, we detail the circuit-level modifications and CAD support necessary to implement the proposed leakage power reduction technique.

4.1 Pulsed-Signalling Circuit

Fig. 6 shows the proposed routing buffer, and as depicted, the buffer can loosely be divided into four stages: receiver stage, gating stage, output stage, and low-leakage keeper stage. Starting at the receiver stage, we note that in contrast to conventional FPGA routing buffers whose first stage consists of an inverter with a level-restoring PMOS transistor connected in feedback, the first stage of our buffer consists of

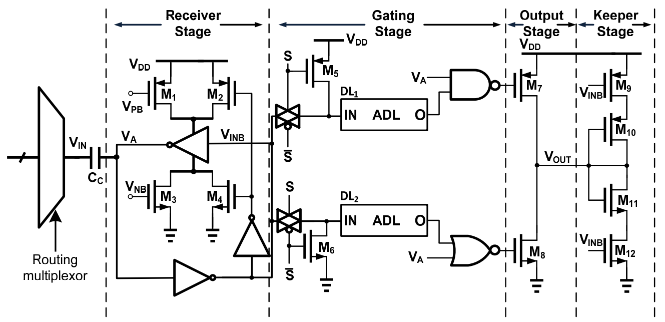


Figure 6: Pulsed-signalling buffer.

an asynchronous fully-regenerative receiver. A regenerative receiver (which is effectively a latch) is made necessary because routing conductors upstream from the routing buffer exist in a low-leakage mode after transmission of data transitions, and as will be made clear later on in this section, the voltage of the routing conductors will tend to drift away from V_{DD} or GND. As such, the burden of retaining state is on the receiver, thus the need for it to be fully regenerative. To eliminate leakage paths, we propose to capacitively couple the output of the routing multiplexer to the input of the reciever with a coupling capacitor C_C; this bears some resemblance to previously proposed AC-coupled on-chip interconnect [2], although previous techniques placed the coupling capacitor at the output of a routing driver in an effort to improve performance and reduce signal swing (and thus power). While many options exist for the implementation of C_C, we propose to use a metal-insulator-metal (MIM) capacitor due to its high capacitance density ($\sim$1-10 fF/μm^2), low series resistance, and the fact that additional active area is not generally required with this option. In a 65nm process, we believe a 5fF MIM cap will be able to fit within the footprint of each routing multiplexer, thus will not requiring any additional area, and can be accomodated with low impact to overall layout and cost.

While a 5fF capacitor is large enough to allow for sufficient signal swing when driving the parasitic capacitance at node V_A, we must ensure that the output impedance of the feedback path of the regenerative receiver is very high, otherwise, all of the charge injected into V_A by C_C will be immediately dissipated. On the other hand, the regeneration time (and thus delay) of the receiver will vary in an inverse manner to the output impedance of the feedback path. To balance these two requirements, a regenerative receiver with hysteresis is employed; we optimized the transistor sizing of this topology to allow for reliable data transmission assuming 6σ variation in transistor parameters (specifically V_T) and $\pm$10% varation in the capacitance of C_C.

Immediately following a rising (or falling) transition at V_{INB}, the output stage is activated with $M8$ ($M7$) turning on while $M7$ ($M8$) remains off, allowing for the output V_{OUT} to fall (rise). Observe that immediately following a transition, the outputs of the (non-inverting) delay lines DL_1 and DL_2 hold the *previous* value of V_{INB}. Once the rising (falling) transition has made its way through DL_1 and DL_2, $M8$ ($M7$) is shut off, which leaves V_{OUT} tristated. As such, for the duration of time following an input transition when the output of DL_1/DL_2 holds its previous value, the output stage is activated and the input transition is allowed to propagate to the output of the buffer. However, once the outputs of both delay-lines have transitioned to their new states, the output stage is tristated. This therefore ensures a low-impedance path to either rail when data is being transmitted by the buffer, but a high effective output impedance when the buffer is in a quiescent state. One subtle point to note is that DL_1 and DL_2 have assymetric delays: DL_1 has a fast rise delay of τ_f (which in our design was around 25ps), and a slow fall delay of τ_s (in our design, this was set to 300 ps), while DL_2 has a slow rise delay of τ_s and a fast fall delay of τ_f. These two assymetric delay lines are needed to ensure proper functionality in the event of a glitch at the input, as it can be shown that having a single symmetric delay line with equal rise and fall delays τ_D can lead to improper function if a glitch of pulse-width less that τ_D were to appear at the input to the buffer. While very specific values have been indicated here, the routing buffer presented in this work is able to tolerate significant variation in τ_s and τ_f. All that is required is that τ_f is sufficiently smaller than the output transition time at V_{OUT}, which in our experiments was around 180 ps when the buffer is loaded by a length-4 wire, while τ_s must be sufficiently larger.

Transistors $M9$-$M12$ realize weak diode keepers which are needed to bound the DC wander at V_{OUT} when the buffer is dynamically tristated, as excessive DC wander can lead to errors when transmitting data. This is because downstream receivers are capacitively coupled, and as such, the input transitions that are seen by receivers is proportional to the difference between $(V_{DD} - GND)$ and the DC wander. Excessive DC wander may diminish the amplitude of signal transitions seen by downstream receivers to the point where signal transisitons are no longer propagated. The diode keepers are sized to bound the maximum DC wander such that under 6σ parameter variation, the worst case signal transition seen at downstream receivers can still be reliably propagated. Moreover, the use of diode weak keepers ensures that leakage paths to either rail are still high impedance.

Finally, an SRAM cell (not shown) outputs signal S and $\overline{S}$ to set the state of the routing buffer; when S is logic-1, the routing buffer operates in pulsed-signalling mode, allowing us to reduce leakage power, while when the output of S is logic-0, the buffer operates in conventional mode. This SRAM cell is configured when the device is being programmed, and the mode of operation of each routing buffer is determined by an analsyis of the worst case injected noise at the buffer output at compile time. We discuss the impact of noise in the following section.

All circuits were designed and simulated with commercial 65nm STMicroelectronics models for functionality verification and power and timing characterization. The dynamic and leakage power overheads arising from the circuitry needed for pulsed-signalling were negligible compared to the leakage and dynamic power dissipated for a length-4 routing wire with the routing multiplexer fanout arising from a routing architecture consisting of fully populated, Wilton-style switch blocks [14]. On the other hand, there is a small delay penalty when a buffer is used in pulsed-signalling mode, since the inherent DC wander at the output of the buffer will degrade the signal swing seen by downstream routing buffers. Under worst-case conditions (a combination of worst case transistor variation resulting in degraded receiver sensitivity and worst-case DC wander), the delay penalty due to reduced signal swing is approximately 80ps. When the buffer is not used in pulsed-signalling mode, the delay penalty is negligible. As such, in addition to noise considerations discussed below, only conductors on paths with sufficient slack may be used in pulsed-signalling mode.

4.2 Noise Considerations

One potential source of concern with the proposed routing buffer is the fact that since the output is tristated, the buffer is sensitive to crosstalk. With V_{OUT} tristated, capacitive coupling noise from adjacent routing conductors can lead to errors. If the noise exceeds a certain threshold, it may be interpretted as an actual signal transition by downstream routing buffers, since they too are capacitively coupled, and potentially have difficulty distinguishing between noise and actual data. Even if the noise injected into V_{OUT} does not lead to an erroneous data transition on downstream routing buffers, the noise is somewhat *residual*, since by design there are no low impedance paths to either rail, as such the injected noise will decay at a slow rate. This is a problem as it can lead to a degraded amplitude of valid data transitions at V_{OUT} seen at downstream receivers. In the worst case, the amplitude of such transitions is proportional to $(V_{DD} - GND) - (|N_{INJ}^{(max)}| + |DCW^{(max)}|)$ where $N_{INJ}^{(max)}$ is the worst case noise injected onto V_{OUT} and DCW_{MAX} is the worst case DC wander. Therefore, for a given design, this power reduction technique cannot be applied to all routing conductors in the circuit in general, but rather only to those routing conductors whose injected noise can be bounded below some maximum threshold level, which in this paper we call N_S (which for the sake of simplicity is normalized to V_{DD}). Let $f_{MUX}(V)$ represent the voltage at the output of a routing multiplexer whose select input is at V volts. Let V_{RT} represent the minimum voltage amplitude that can be reliably detected by the proposed hysteretic regenerative receiver in the presence of 6σ parameter variation of its constituent transistors, and let V_{RR} represent the maximum voltage amplitude that can be reliably rejected by the receiver, again subject to transistor variation. We therefore require $f_{MUX}(N_S) < V_{RR}$ and $f_{MUX}(V_{DD} - GND - N_S - DCW^{(max)}) \geq V_{RT}$. Through transistor sizing optimization, we found that $N_S \approx 1/4$, given that the routing multiplexers use a boosted gate voltage, which is a commmon feature in commercial FPGAs [1]. For 65nm, the maximum reliable gate voltage is 1.2 V.

Let $C(i,j)$ represent the coupling capacitance between conductors i and j, $C(i)$ the total (plate and coupling) capacitance of conductor i, and $Adj(i)$ the set conductors within the same metal layer and adjacent to i. Also, let $occ(i)$ be equal to 1 if i is a used conductor or 0 if i is unused. Therefore, for conductor i to operate in pulsed-signalling mode, the following condition must hold:

$$\sum_{j \in Adj(i)} \frac{C(i,j)}{C(i)} occ(j) \leq N_S \tag{2}$$

In the following sections, we discuss how a conventional FPGA CAD flow may be modified to optimize leakage power by maximizing the number of used conductors which are to operate in pulsed-signalling mode, subject to the constraints described in this section.

5. CAD SUPPORT

As mentioned previously, all three power reduction techniques considered in this paper leverage the fact that FPGAs often have many unused routing conductors, and by ensuring specific routing conductors are unused, different power reduction opportunities are available for each of the power reduction techniques. For the remainder of this paper, we define $F(n)$ to be a set of "friend" conductors of n: routing conductors whose occupancies determine the power reduction opportunities for conductor n. For the charge recycling technique, $F(n)$ will always contain a single element, the paired routing conductor of conductor n. For the capacitance optimization and pulse-signalling techniques, $F(n)$ is the set of conductors adjacent to n within the same metal layer. The power reduction opportunities for each of the aforementioned techniques can then be expressed as follows: The effective capacitance of conductor n is proportional to the number of conductors $j \in F(n)$ which are used (and their respective individual coupling capacitances with conductor n), thus we seek to minimize this quantity to reduce the dynamic power of conductor n when using the capacitance optimization technique described in Section 3.1. For the charge recycling technique described in Section 3.2, if n is a used routing conductor, we seek to ensure the sole member of $F(n)$ is unused so that it may be used as a charge reservoir. Finally, to minimize active leakage power of a routing conductor using the technique described in Section 4, we need to ensure that a *sufficient* number of conductors $j \in F(n)$ are unused, and thus can be used as shielding conductors to allow n to safely operate in pulsed-signalling mode, as described in Section 4.2.

As such for each technique, special effort must be made during the physical implementation of a circuit to ensure that while a signal is routed between source and sink pins, specific routing conductors along the path are unused in order to minimize power. Our proposed CAD flow builds on VTR (a conventional FPGA CAD flow) [11], where we have made modifications to the router and added a routing conductor mode selection phase which takes place after routing and timing analysis.These modifications are discussed in the following sections.

5.1 Power-Aware Router

The VPR router incorporates the PathFinder algorithm [9] which was discussed in Section 2. The cost function incorporates metrics of timing performance and *routability* (i.e. the ability to legally route all of the connections in the circuit) in order to yield a routed circuit with a favorable quality of results. The cost function implemented VPR is:

$$\begin{aligned} Cost_n \quad &= \quad (1 - Crit_i) \cdot cong\_cost_n \\ &\quad + Crit_i \cdot delay\_cost_n \end{aligned} \tag{3}$$

where n is the routing resource (e.g. wire segment) being considered for addition to a partially-completed route, i is the driver/load connection being routed, $Crit_i$ is the timing criticality of the connection (equal to 1 for connections on the critical path, and decreasing to 0 as the slack of the connection increases), $cong\_cost_n$ is the congestion cost of routing resource n which gives an indication of the *demand* for the routing resource among nets, while $delay\_cost_n$ is the delay of routing resource n. The reader is referred to [9] for further discussion on this cost function.

Turning now to our modification of the VPR router, for a routing segment, n, we use the following cost function:

$$\begin{aligned} Cost_n \quad &= \quad (1 - Crit_i) \cdot [cong\_cost_n \\ &\quad + PF \cdot power\_cost_n \\ &\quad + AF \cdot power\_cost\_infringe_n] \\ &\quad + Crit_i \cdot delay\_cost_n \end{aligned} \tag{4}$$

where PF and AF are empirically determined scalar weighting terms, $power\_cost_n$ is a term to guide power critical nets to use routing conductors which are likely to operate in a low-power mode, and $power\_cost\_infringe_n$ is used to guide nets *away* from the "friend" conductors of a potentially used routing conductor which is likely to operate in lower power mode. For the sake of brevity, in the remainder of this section we will focus on the form of these two terms as they relate to the pulsed-signalling power reduction technique; for a detailed description of the router cost functions for the charge recycling and capacitance optimization techniques, the reader is referred to [3] and [4], respectively. For the pulsed-signalling power reduction technique, the term $power\_cost_n$ is given by:

$$power\_cost_n = \begin{cases} 1 & : \sum\limits_{j \in F(n)} \frac{C(n,j)}{C(n)} occ(j) - N_S > 0 \\ 0 & : otherwise \end{cases}$$

where $C(n,j)$, $C(n)$, $occ(j)$, and N_S are as defined in Section 4.2. The term $power\_cost\_infringe_n$ is equal to:

$$\sum_{j \in F(n)} \frac{C(n,j)}{\sum\limits_{k \in F(j)} C(j,k) occ(k)} \cdot occ(j) \cdot power\_cost_j \cdot (1 - max\_crit_j) \tag{5}$$

where max_crit_j is the worst case criticality over all nets currently vying for j.

The motivation for the modification to the cost function is as follows: while routing a circuit, we wish to maximize the number of active conductors which can be put into pulsed-signalling state in order to reduce leakage power. In general, when routing connection i, we need to consider two cases: 1) connection i has sufficient timing margin, or 2) connection i is timing critical. For the former case, we strive to optimize the routing for two different situations. First, we attempt to guide nets to use segments with a sufficient number of unoccupied neighbours; the term $power\_cost_n$ attempts to optimize for this goal, since it estimates whether or not the noise on conductor n exceeds N_S. Any routing conductor n whose $power\_cost_n$ is equal to 0 may be used in pulsed-signalling mode. Conversely, if $power\_cost_n$ is greater than 0, it is unlikely to operate in pulsed-signalling mode, therefore we attempt to avoid using it. The second goal is to avoid injecting excessive noise onto tracks which may otherwise potentially be used in pulsed-signalling mode. This optimization goal is dealt with by the term $power\_cost\_infringe_n$: in evaluating conductor n, we examine all neighbours j of n which are used, and assess the likelihood of j operating in pulsed-signalling mode. If j is likely to be used by a low criticality net (given by the 1 - max_crit_j term) and yet has $power\_cost_j > 0$ (i.e. there was an opportunity for j to operate in pulsed-signalling mode and yet it cannot due to excessive noise), we should assign some cost to using n. This cost is scaled by $C(n,j) / \sum\limits_{k \in F(j)} C(j,k) occ(k)$, since the cost of j not operating in pulsed-signalling mode should be distributed over its neighbours and weighted according to the noise-injected by each of j's neighbours. Finally, if the current net has high criticality, then it is unlikely that i can be used in pulsed-signalling mode due to its inherent delay penalty, and as such we opt to focus on optimizing i's delay.

5.2 Routing Wire Mode Selection

After routing and timing analysis have completed, we are in a position to determine the operating modes of the routing conductors in the chip. For the capacitance optimization technique, mode selection is trivial, since all unused conductors are automatically tristated. For the charge recycling power reduction technique, we formulated the mode selection problem, where power is to be optimized subject to timing constraints, as a mixed integer linear program (MILP). The reader is referred to [3] for further details. For the pulsed-signalling technique presented in this work, we again seek to optimize (leakage) power, subject to timing and maximum injected noise constraints. Algorithm 1 provides pseudocode for the greedy approach we used in this study to solve this problem. With the core objective of minimizing leakage power, the goals of the algorithm are to determine which of the used routing conductors will operate in pulsed-signalling mode, which of the unused routing conducts will act as *shields* to help isolate used conductors operating in pulsed-signalling mode from adjacent noise sources, and which of the unused conductors can be set in low-leakage, tristated mode. In the pseudocode shown $noise(i)$ is the worst case total injected noise on conductor i, and is equal to $\sum_{j \in F(i)} occ(j) \cdot C(i,j)/C(i)$, $selects(i)$ is the number of MUXes in the fanout of i where i connects to a selected input pin, $slack(G,i)$ returns the timing slack available for the connection associated with routing conductor i given timing graph G, $\mathrm{update\_timing\_graph(G,i,t)}$ is a routine (not shown for the sake of brevity) which *incrementally* updates the timing graph G when the delay of the connection associated with routing conductor i is increased by t seconds (i.e. the routine only updates edges that are affected, and avoids a complete timing analysis). Recall that t_D is the delay penalty of operating in pulsed-signalling mode (as mentioned previously, this is $80ps$), and the other terms are as described previously. It should be noted that since it can be shown that typically the input pins of a routing multiplexer which have the greatest leakage current are the selected input pins (the pass-transistors

Algorithm 1: Routing wire mode selection

```
Input: a set $W$ of routing conductors, a timing graph $G(V,E)$
Output: a data structure $states(i)$ mapping routing conductor $i$ to its state
1  begin
2      $states(i) \leftarrow unknown$ for all $i \in W$
3      /* Build the set of used wires */
4      $WU \leftarrow \{x \mid x \in W \wedge occ(x) = 1\}$
5      /* Build the set of unused wires */
6      $WN \leftarrow \{x \mid x \in W \wedge occ(x) = 0\}$
7      /* First determine the state of used conductors */
       foreach $i \in WU$ in descending order of selects(i) do
8          if $noise(i) > N_S$ or $slack(G,i) < t_D$ then
9              $states(i) \leftarrow ungated$
10         else
11             $states(i) \leftarrow gated$
12             /* Update G to reflect state of $i$ */
13             update_timing_graph$(G, i, t_D)$
14             /* Compute noise margin */
15             $nm \leftarrow noise(i) - N_S$
16             foreach $j \in F(i) \cap WN$ in descending order of selects(j) do
17                 if $C(i,j)/C(i) < nm$ and $states(j) == unknown$ then
18                     /* conductor $j$ can potentially be gated, don't set its state yet */
19                     $nm \leftarrow nm - C(i,j)/C(i)$
20                 else
21                     /* use $j$ as a shield */
22                     $states(j) \leftarrow ungated$
23                 end
24             end
25         end
26     end
27     /* Now finalize the states of unused conductors */
       foreach $i \in WN$ do
28         if $states(i) == unknown$ then
29             $states(i) \leftarrow gated$
30         end
31     end
32 end
```

connected to these pins are "on", thus these pins have low impedance), the worst case leakage power of a routing conductor is approximately proportional to $selects(i)$. As such, we use this quantity as an estimate for leakage power where appropriate in the algorithm.

The algorithm takes as input the timing graph for the circuit being optimized, and a set W for all the used and unused routing wires in the circuit. The output of the algorithm is the *state* for each routing conductor i, and is equal to either *gated* or *ungated*. A used routing conductor will operate in pulsed-signalling mode if its state is *gated*, and will operate in conventional mode if its state is *ungated*. An unused routing conductor will be tristated if its state is *gated*, and will act as a shield if its state is *ungated*. While we wish to maximize the number of used conductors that are operating in pulsed signalling mode, we also wish to maximize the number of unused conductors in tristate mode, as they lead to reduced leakage.

The first part of the algorithm determines the state of all the used routing conductors in the circuit. We visit these conductors in descending order of their estimated leakage power (using $selects(i)$ as an estimate). For each routing conductor i, we assess if either the injected noise onto i is greater than N_S or if sufficient timing slack is unavailable, in which case i cannot be used in pulsed-signalling mode, and therefore its state is set to *ungated*. Otherwise, i can be used in pulsed-signalling mode and will have its state set to *gated*, whereafter G is updated by increasing the delay of the connection associated with conductor i by t_D seconds. Following this, we compute the *noise margin* for conductor i, denoted nm in the pseudocode, which is the difference between the total noise injected onto i and N_S. If this is non-zero, then *some* of the unused neighbouring conductors of i may b tristated, leading to reduced leakage. We determine which of the unused conductors $j \in F(i)$ may be potentially tristated by visiting them in descending order of their estimate leakage power (again using $selects(j)$ as an estimate), and then assessing if the injected noise onto i from j is less than the current value of nm. If this quantity is less than nm, then j may potentially be tristated, however we leave its state as *ungated*, and revisit the conductor after all used conductors have been visited. If instead the noise injected from j onto i is greater than nm we set the state of j to *ungated*, and thus j will act as a shield.

After the states of all used conductors are determined, we visit each unused conductor i whose state is still set to *unknown*. These conductors are either not adjacent to any used conductors, or the determination of their state was deferred until this phase of the algorithm. This would only occur when an unused conductor is adjacent to ungated used conductors and/or gated used conductors with sufficient noise margin. As such, all of these conductors can be tristated, meaning $states(i) = gated$.

While the above approach is greedy, as will be discussed in the next section, our results appear to indicate that it works well. Nonetheless a less greedy approach will likely improve the quality of results further, but investigation of such approaches is left for future work.

6. EXPERIMENTAL STUDY

In order to assess the merits of the proposed power reduction techniques, we used the set of benchmark circuits packaged with VTR [11]. Our baseline architecture contains unidirectional wire segments which span 4 CLB tiles, uses the Wilton switch block [14], and has logic blocks with ten 6-LUTs/FFs per CLB. All benchmark circuits were routed on the baseline architecture to determine the minimum number of tracks per channel needed to route each circuit successfully (W_{min}); all routing buffers in the baseline architecture are conventional. For each circuit, we then computed $W = 1.3 \times W_{min}$ to reflect a medium-stress routing scenario – standard approach in FPGA architecture research. The computed W value for each circuit was used for all experimental runs of the circuit. We used the ACE switching activity estimator tool [7] to compute switching activity for each signal in each benchmark circuit, as this information is needed for the charge recycling and capacitance optimization techniques. For the pulsed-signalling technique, we consider an additional set of experiments. FPGA vendors develop FPGA *families* (such as Stratix from Altera or Virtex from Xilinx) which are a set of different sized FPGAs all having the same fundamental routing and logic architecture, but have different numbers of CLBs, hard blocks, I/O capabilities, etc. While the different members of a family may have vastly differing logic capacities, their routing architecture (and thus channel width) remains constant; this is a practical approach since it allows a vendor to allocate design resources to develop a single tile (for example containing a single CLB and associated routing channels), and then stitch a varying number of these tiles together to form the different members of each family. As such, a more realistic scenario is to assume a *fixed channel width*, W_{max}, which we set to be 10% larger than the largest channel width for the benchmarks considered in this study. Thus, for the pulsed-signalling technique, we experiment with with $W = 1.3 \times W_{min}$ and $W = W_{max}$.

In our experiments, we assess the interconnect dynamic power reduction, static power reduction, and area overheads. For each of the techniques, the associated router cost functions contain scalar terms which were optimized, however for the sake of brevity we exclude details and the optimization of these terms (the reader is referred to [3] and [4] for more detail). The results presented here therefore reflect power reductions obtained with optimized cost functions; a circuit-by-circuit breakdown of our results is provided in Table 1. The critical path degradations were negligible for all three power reduction techniques, thus for the remainder of this section, we focus solely on power reduction and area overhead results.

6.1 Power Reduction

For the capacitance optimization technique, dynamic power reduction ranges from 10 to 19% with a geometric mean of 13.5 %; this is assuming the ratio between coupling capacitance (C_C) and "plate" capacitance (C_P) is 2; greater power reductions are possible as this ratio increases (see [4] for further discussion). This is in addition to a reduction in interconnect static power of approximately 15.2% (geometric mean across benchmarks). The charge recycling technqiue offers similar dynamic power reduction, ranging from 13 to 19% with a geometric mean of just under 16%.

Turning now to the reductions afforded by the proposed pulsed-signalling technique, we see that active leakage (i.e. leakage in the used routing conductors) can by 26% on average when a circuit is targetted towards an FPGA with channel width $1.3 \times W_{min}$, while total leakage in the routing network (including leakage from unused conductors) can be reduced by approximately 31%. For the fixed channel width experiments (circuits target an architecture with channel width of W_{max}), the active leakage reduction dramatically increases to ∼62% on average, however the increase in total routing leakage reduction is less dramatic as in this case it is now ∼37% on average. The justification for these results are as follows: since for most circuits in our benchmark set, $1.3 \times W_{min}$ is *significantly* less than W_{max} (due to the fact that the logic counts for the designs in the benchmarks may vary by over an order of magnitude, in addition to varying levels of routing complexity), when routing designs at

Table 1: Power reduction and area overhead results.

	Effective Capacitance Optimization			**Charge Recycling**		**Pulsed-signalling**					
						W = 1.3 x Wmin			**W = Wmax**		
Circuit	Routing Dynamic Power Reduction [%]	Routing Static Power Reduction [%]	Routing Area Overhead [%]	Routing Dynamic Power Reduction [%]	Routing Area Overhead [%]	Active routing leakage reduction [%]	Total routing leakage reduction [%]	Routing Area overhead [%]	Active routing leakage reduction [%]	Total routing leakage reduction [%]	Routing Area overhead [%]
blob_merge	10.4	13.2	4.7	13.6	24.8	26.0	31.5	20.9	61.7	37.6	20
boundtop	13.3	13.6	4.7	16.5	24.5	25.5	31.4	20.7	69.8	40.0	20
ch_intrinsics	19.2	15.5	5.1	17	27	24.6	31.1	22.8	64.2	36.0	20
diffeq1	16.4	16.7	5.1	17.4	26.8	24.7	30.3	22.6	60.5	36.3	20
diffeq2	17.6	18	4.9	13.3	25.7	23.5	28.5	21.7	60.3	38.0	20
LU8PEEng	10.6	12.7	4.6	15.9	24	23.3	31.5	20.3	51.8	35.6	20
mkDelayWorker32B	13.2	18.1	4.5	16.5	23.8	25.2	30.0	20.1	62.0	36.6	20
mkPktMerge	16.5	21	4.4	12.9	22.9	27.5	31.5	19.3	62.7	39.4	20
mkSMAdapter4b	12.4	15.2	4.7	16.5	24.7	24.8	30.6	20.9	59.6	36.0	20
or1200	12.4	16.3	4.6	18.7	24.1	26.0	30.6	20.4	61.8	36.7	20
raygentop	12.9	13.9	4.8	15.8	25.1	28.0	31.2	21.2	60.0	37.9	20
sha	11	13.3	4.7	15.9	24.8	25.5	31.0	20.9	67.9	40.0	20
stereovision0	12	13.8	4.4	16.6	23.1	27.0	32.0	19.5	69.5	37.4	20
stereovision1	11	14.3	4.6	14.2	24.2	29.0	30.8	20.4	74.0	38.0	20
stereovision2	12.9	14.1	4.6	14.9	24.1	31.8	30.8	20.4	48.0	34.2	20
stereovision3	19.2	15.8	6.2	15.1	32.4	25.2	30.5	27.4	53.0	34.7	20
mean	**13.8**	**15.3**	**4.8**	**15.7**	**25.1**	**26.1**	**30.8**	**21.2**	**61.7**	**37.2**	**20.0**

W_{max}, the increased number of unused routing conductors means that the used routing conductors may more easily be spaced apart from one another when nets are being routed. This explains why the active leakage reduction increased so dramatically. On the other hand, a larger channel width means that the ratio between unused conductors to used conductors increases, meaning that the net impact on total routing leakage from used routing conductors is diminished. This explains why while active leakage reduction increased almost three-fold going from $W = 1.3 \times W_{min}$ to $W = W_{max}$, total routing leakage reduction increased by approximately $\sim$21%.

6.2 Area Overhead

Of the three power reduction techniques, the capacitance optimization technique offers the lowest (routing) area overhead of just under 5% on average. Given that for commercial architectures routing area is approximately 50% of total FPGA core area [8], this means that total core FPGA area grows by under 2.5%. The routing area overheads for the charge recycling and pulsed-signalling are similar at 25% and $\sim$20%, respectively, which translates to a total core area overhead of just over $\sim$10% for both techniques.

It is important to consider the impact which these area increases will have on routing power. For the capacitance optimization technique, a $\sim$2% increase in tile area corresponds to $\sqrt{1.02} \approx 1.01\times$ increase in tile dimensions, meaning that increase in dynamic power resulting from the increased tile size is less than 1%. Similarly, the charge recycling and pulsed-signalling techniques will result in a $\sqrt{1.1} \approx 1.05\times$ increase in tile dimensions. For the charge recycling technique, this effective power overhead is less than the power saved, although further optimization of the circuitry and/or technique is required for the power savings to be more meaningful. On the other hand, given the significant power savings from the pulse-signalling technique, a $\sim$5% increase in routing dynamic power for a 30-40% decrease in routing leakage power is a favourable tradeoff. Assuming static power is 1/3 dynamic power, these numbers indicate that total routing power can still be reduced by $\sim$10%. For many designs that run at lower clock rates, the dynamic power overhead will decrease, while the leakage power reduction will stay the same, meaning that total routing energy savings will increase.

7. CONCLUSIONS AND FUTURE WORK

This paper presented three novel techniques to reduce power consumption in FPGA interconnect. The techniques showed the interconnect dynamic power can be reduced by 10-20%, while leakage power can be reduced by almost 40%. Because of the fact that all of the techniques fundamentally require the same set of conditions to reduce power, namely that routing conductors which consume a lot of power (i.e. if used by a net with high switching activity and/or connected to many selected input pins of downstream routing multiplexers), and because the circuitry required for the techniques are all similar to one another, it is likely that these techniques may be combined to allow for signiciant net decrease in dynamic and leakage power consumption in FPGA interconnect. As assessment of the net benefits in combining a set of these techniques is an interesting avenue for future research. In addition, future work will aim to improve CAD tools to improve quality of results and reduce tool runtime.

8. REFERENCES

[1] C. Chiasson and V. Betz. Should FPGAs abandon the pass-gate? In *IEEE FPL 2013*.

[2] R. Ho et al. High speed and low energy capacitively driven on-chip wires. *IEEE JSSC*, 43(1):52–60, Jan 2008.

[3] S. Huda et al. Charge recycling for power reduction in FPGA interconnect. In *IEEE FPL 2013*.

[4] S. Huda et al. Optimizing effective interconnect capacitance for FPGA power reduction. In *ACM/SIGDA FPGA 2014*.

[5] ITRS. ITRS 2011 Report.

[6] I. Kuon and J. Rose. Measuring the gap between FPGAs and ASICs. *IEEE TCAD*, 26(2):203–215, Feb. 2007.

[7] J. Lamoureux and S. Wilton. Activity estimation for field-programmable gate arrays. In *IEEE FPL 2006*.

[8] D. Lewis et al. Architectural enhancements in Stratix V$^{\text{TM}}$. In *ACM/SIGDA FPGA 2013*.

[9] L. McMurchie and C. Ebeling. Pathfinder: A negotiation-based performance-driven router for FPGAs. In *ACM/SIGDA FPGA 1995*.

[10] A. Putnam et al. A reconfigurable fabric for accelerating large-scale datacenter services. In *ACM/IEEE ISCA 2014*.

[11] J. Rose et al. The VTR project: Architecture and CAD for FPGAs from verilog to routing. In *ACM/SIGDA FPGA 2012*.

[12] L. Shang et al. Dynamic power consumption of the Virtex-II FPGA family. In *ACM/SIGDA FPGA 2002*, 2002.

[13] T. Tuan and B. Lai. Leakage power analysis of a 90nm FPGA. In *IEEE CICC 2003*.

[14] S. J. Wilton. *Architecture and algorithms for field-programmable gate arrays with embedded memory.* PhD thesis, 1997.