Dynamic Idle Core Management and Leakage Current Reuse in MPSoC Platforms

MD Shazzad Hossain and Ioannis Savidis
Drexel University, PA, USA

ABSTRACT

In this paper, algorithmic and circuit techniques are proposed for dynamic power management that allows for the reuse of the leakage current of idle circuit blocks and cores in a multiprocessor system-on-chip platform. First, a novel scheduling algorithm, *longest idle time - leakage reuse (LIT-LR)*, is proposed for energy efficient reuse of leakage current, which generates a supply voltage of 340 mV with less than ±3% variation across the `tt`, `ff`, and `ss` process corners. The *LIT-LR* algorithm reduces the energy consumption of the leakage control blocks and the peak power consumption by, respectively, 25% and 7.4% as compared to random assignment of idle cores for leakage reuse. Second, a novel usage ranking based algorithm, *longest idle time - simultaneous leakage reuse and power gating (LIT-LRPG)*, is proposed for simultaneous implementation of power gating and leakage reuse. Applying power gating with leakage reuse reduces the total energy consumption of the MPSoC by 50.2%, 14.4%, and 5.7% as compared to, respectively, a baseline topology that includes neither leakage reuse or power gating, only includes power gating, and only includes leakage reuse.

1 INTRODUCTION

Energy loss due to leakage current is inevitable in modern processors and system-on-chips (SoCs), which results in energy waste as no computation or data storage is performed with leakage current. The state-of-the-art CPU, GPU, DNN accelerators, SoCs, and battery operated devices consume a significant portion of the total energy as leakage due to an increased number of on-chip computation and memory units as well as increased idle times while executing a diverse set of applications [1–4].

Power gating (PG), a method to disconnect idle circuits from the power network, is a widely used power management technique to reduce the leakage current [1, 3, 5–10]. Techniques developed through prior research apply cut-off and intermediate mode power gating at the core, block, memory, and network-on-chip (NoC) level [1, 4, 6, 8–11]. Power gating with single and multiple sleep modes raises the voltage at the virtual ground node by up to VDD when applying a power-down mode and, therefore, reduces the leakage current of the circuit [6, 7, 11]. However, the reduction in the leakage current is achieved at a cost of increased current transients on the power distribution network (PDN) and a large power consumption by the footer transistors (MOS switches) due to the discharging of charge on the virtual ground node to true GND during mode transitions [5, 7].

Reusing leakage current from idle circuits provides significant improvement in the overall energy efficiency while also generating intrinsic voltage source(s) for ICs with multiple voltage domains [12, 13]. In this paper, both circuit and algorithmic techniques are proposed to manage and reuse the leakage current of a circuit such that the overall energy consumption in a multi-processor system-on-chip (MPSoC) is reduced. In addition, the simultaneous implementation of the leakage reuse (LR) and power gating techniques is explored to further improve the energy efficiency. A controlled reuse (recycling) of the leakage current of an idle core generates a virtual ground voltage *VGND* that is used as a supply voltage for an active core. In addition, the leakage reuse technique reduces the leakage current through the idle cores. The proposed technique provides two primary benefits: 1) A reduction in leakage current without a large overhead in wake-up energy due to the inherent voltage-stacking effect, and 2) an improvement in the overall energy efficiency of the IC by operating active core(s) with recycled leakage charge.

In this paper, idle cores from which leakage current is reused are described as *donor* cores and active cores to which reused charge is delivered as *receiver* cores. The proposed power management techniques are evaluated through a circuit model simulated in SPICE to accurately characterize the improved energy efficiency and the dynamic transients on the power delivery network (PDN) while accounting for process variation.

The primary contributions of this paper include

- a novel dynamic idle core management technique and scheduling algorithm, *longest idle time - leakage current reuse (LIT-LR)*, to optimally assign the idle donor circuit block(s) or core(s) for leakage current reuse, and
- a novel technique and algorithm, *longest idle time - simultaneous leakage reuse and power gating (LIT-LRPG)*, to optimally assign idle cores for simultaneous execution of power gating and leakage reuse.

The rest of the paper is organized as follows. The proposed leakage current reuse technique is discussed in Section 2.1 and an algorithm to efficiently schedule the idle cores for leakage reuse is described in Section 2.2. The simultaneous implementation of leakage reuse and power gating is discussed in Section 3. The simulation framework and corresponding results are discussed in, respectively, Section 4 and Section 5. Some concluding remarks are provided in Section 6.

2 LEAKAGE CURRENT REUSE

The proposed power management technique is applied to only idle cores such that the circuit is unaffected when operating in active mode. The leakage current from idle cores is reused and delivered to an active core within a MPSoC through voltage stacking [14].

2.1 Circuit and System Model of Leakage Reuse

The proposed circuit model that accounts for the reuse of leakage current is shown in Fig. 1. In a MPSoC platform, m number of donor cores are connected to k number of receiver cores through m number of *leakage control blocks (LCBs)*, which consists of two transistors behaving as switches. Each of m number of LCB is implemented with an n-bit (n is 1 for Algorithm 1 and 2 for Algorithm 2) control signal Φ such that Φ_1 is applied to LCB_1, Φ_2 to LCB_2, and Φ_m to LCB_m. Each control signal $\Phi_1, \Phi_2, \dots\ \Phi_m$ is connected to the gate terminals of the S_{VG} and S_G transistor of the corresponding LCB_m. Note that both the number of donor cores and receiver cores are scalable at design time. Under normal circuit activity, when a donor core $D_{CORE,m}$ is executing a workload, Φ_m is set to logic *high* and is applied to the LCB_m such that the ground node of all transistors within the donor core $D_{CORE,m}$ is connected to true ground (*GND*). The normal activity of the donor cores, therefore, remains unaffected.

ISLPED '20, August 10–12, 2020, Boston, MA, USA
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7053-0/20/08…$15.00
https://doi.org/10.1145/3370748.3406558

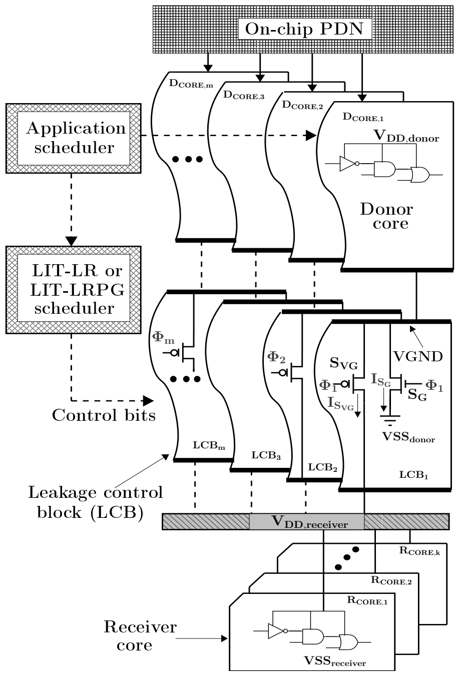


Figure 1: System level model of the leakage reuse technique with dynamic idle core assignment.

At the onset of an idle period, where the energy conserved exceeds the break-even point as defined by $N_{BE,LR,m}$ in Algorithm 1, Φ_m is set to logic *low* and is applied to LCB_m such that the donor core $D_{CORE,m}$ and the receiver core $R_{CORE,k}$ are stacked. The stacking serves two primary purposes: 1) to produce a virtual ground voltage that is used as the supply voltage of the receiver core $R_{CORE,k}$ and 2) to reduce the total leakage current of the idle donor core $D_{CORE,m}$. The conventional voltage stacking technique stacks two cores (a top PDN at a higher potential and a bottom PDN at a lower potential) regardless of the circuit activity during both active and idle periods, where the executing workloads of the top core affects the supply voltage of the bottom core [14]. Therefore, expensive on-chip voltage regulators and closed-loop power management techniques are required to maintain a steady supply voltage for the bottom core [14]. However, the proposed technique only stacks idle donor cores (top core(s)) with no workload activity with receiver core(s) (bottom core(s)). Expensive voltage regulators are, therefore, not required to maintain a stable supply for the receiver cores. The simulated results presented in Section 5 indicate that the use of only on-chip decoupling capacitors is sufficient to negate the voltage transients on the PDN and, therefore, maintain a stable supply voltage for the receiver core.

The number of donor cores required to generate a desired supply voltage for a given receiver core is determined at design time. The desired supply voltage of a receiver core is set by either adjusting the ratio of the area of the donor and receiver cores, modifying the size of the switch in the LCB, or applying a charge booster such as an on-chip switch capacitor based voltage doubler to increase the supply level of a receiver core. In this paper, the supply voltage is set through the ratio of the areas of the donor and receiver cores.

2.2 Scheduling of Idle Donor Cores for Leakage Current Reuse

Conventional hardware and software based power management techniques do not implement leakage reuse despite the opportunities of efficiently conserving energy from idle donor cores [2, 8, 15]. In this paper, idle donor cores or circuit blocks are selected for leakage reuse to minimize the number of transistors switching in the *leakage control block* and to maximize the energy-efficiency of a MPSoC system. The proposed algorithm is equally applicable to core or circuit block level assignment for leakage reuse.

The proposed technique requires the scheduling information of donor cores to effectively select the idle donor cores for leakage reuse. In order to evaluate the effectiveness of the proposed leakage reuse technique, in the worst case, the application scheduling algorithm executes a task with the shortest duration possible and with a minimal idle time in a multi-core platform. The heterogeneous earliest-finish time (HEFT) algorithm meets both requirements as 1) the overall completion time is minimized by providing an execution rank for each task in an application for a set number of cores in a multi-core system and 2) tasks are scheduled based on an insertion-based policy that assigns tasks in idle time slots between two already-scheduled tasks on a core, which reduces the total idle time of the system [16].

Recently, dynamic voltage and frequency scaling (DVFS) is applied in conjunction with the execution of the HEFT algorithm, which provides an estimate of the total number of processor cycles executed by each core for a given task within a frequency range of 1 GHz to 2 GHz [2]. In this paper, the DVFS enabled HEFT algorithm is used as the application scheduler for the donor cores, which sends the scheduling information to the proposed *longest idle time - leakage current reuse (LIT-LR)* algorithm as shown in Fig 1. The *LIT-LR* algorithm generates and sends the control bit(s) to the LCB when scheduling the idle donor cores for leakage current reuse. In addition, a directed acyclic graph (DAG) is used to represent the application workload of the donor cores and is utilized by the HEFT algorithm.

Algorithm 1 Idle core assignment for leakage current reuse

```
Input: $m$, $Task_{num}$, $Order_{core}$, $Pcycles_{task}$, $N_{BE,LR}$
Output: Available idle core assignment to be used for leakage reuse with usage-based ordering
($LR_{cores}$)
 1: while $i \leq Task_{num}$ do
 2:     Calculate the total number of cycles until the application is assigned to the next processor
          for execution
 3:     if $Pcycles_{task}[i] > N_{BE,LR}$ then ▹ $N_{BE,LR}$ is the number of cycles corresponding to the
          energy break-even point for LR
 4:         while j $\leq (Task_{num} - i)$ do
 5:             Track the earliest execution start time of all of $Task_{num}$ - $i$ tasks in $m$ number
                  of cores.
 6:             Store a ranking of each core in $Core_{m,ranking}$ based on the order in which each task
                  is executed.
 7:         end while
 8:         if $i \neq (Task_{num}$ - 1) then
 9:             $LR_{cores\_index}[i] \Leftarrow max(Core_{1,ranking}, Core_{2,ranking}, Core_{3,ranking}, \dots$
                  $Core_{m,ranking})$ ▹ Assigning the donor core with longest idle time for LR
10:             $LR_{cores}[i] \Leftarrow Order_{core}\left[LR_{cores\_index}[i]\right]$
11:         else if $i == (Task_{num}$ - 1) then
12:             $LR_{cores}i \Leftarrow LR_{cores}[0]$ ▹ Tracking the donor core with longest idle time at the
                  beginning of last execution and assigning for current state
13:         end if
14:     else
15:         $LR_{cores}[i] \Leftarrow Order_{core}\left[LR_{cores\_index}[i-1]\right]$ ▹ Keep the last idle core assignment
          for leakage reuse
16:     end if
17: end while
18: return $LR_{cores}$
```

2.3 Longest Idle Time - Leakage Current Reuse Algorithm

The pseudo-code of the proposed *LIT-LR* algorithm that efficiently schedules idle donor cores for leakage current reuse is provided as Algorithm 1. First, the execution order of each core $Order_{core}$ is calculated for the given task graph using the HEFT algorithm [16]. Then, the number of processor cycles required for each task $Pcycles_{task}$ is calculated in a DVFS enabled MPSoC platform [2]. The calculated values of $Order_{core}$ and $Pcycles_{task}$ are applied as inputs to both Algorithms 1 and 2. Based on the number of cores m, number of tasks in each task graph $Task_{num}$, $Order_{core}$, $Pcycles_{task}$, and number of cycles corresponding to the break-even point $N_{BE,LR}$, the *LIT-LR* algorithm determines the earliest execution time of each task on the m core system. Depending on the task execution order and idle intervals, each core is dynamically assigned a ranking, where the core with the longest idle time is given the highest ranking and the core with shortest idle time is assigned the lowest ranking. The scheduler gives priority to the highest ranking idle donor cores when assigning cores for leakage current reuse and sends the control bit Φ to the corresponding LCB. In this work, a single receiver core is assumed, and one donor core provides sufficient leakage current for the receiver core.

3 SIMULTANEOUS IMPLEMENTATION OF LEAKAGE REUSE AND POWER GATING

Current state-of-the-art power management techniques and algorithms do not include leakage current reuse with power gating despite the opportunity of significant improvement in energy-efficiency. However, despite the potential benefits, the overall system energy efficiency is not improved unless all idle donor cores are scheduled for leakage current reuse at any given time. In a MPSoC platform, the idle donor cores that are not scheduled for leakage current reuse continue to leak current. In this paper, the simultaneous implementation of power gating and leakage reuse is considered, which is applicable at the block, core, memory, and NoC level. Only donor cores are considered for power gating.

The opportunity of applying both power gating and leakage reuse introduces new challenges as both techniques target idle circuits or cores in a multi-core system. Power gating is best utilized when the core with the longest idle time is placed in deep sleep mode and the core with shortest ($\geq$ break-even point $N_{BE,PG}$) idle time is placed in light sleep mode [11]. Note that deep sleep and light sleep correspond to the largest and smallest wake-up penalty, respectively. Similarly, the leakage reuse technique is best utilized when the core with the longest idle time is placed in leakage reuse mode to minimize the energy consumption of switching the LCBs. Therefore, the maximum benefit in energy efficiency is achieved when the cores with the longest idle times are assigned for either power gating or leakage reuse. Assigning the longest idle core for leakage reuse is beneficial only when the donor core provides just the sufficient amount of current to the receiver core. If the amount of supplied leakage current from a particular donor core is more than a receiver core requires, than the energy efficiency is reduced as an undesired increase in the *VGND* voltage occurs. Power gating improves the overall energy efficiency for a given donor core only when the reduction in the leakage current from applying power gating is greater than savings provided by leakage current reuse (reduction in leakage current plus the total energy consumed by the receiver core). Therefore, the assignment of idle cores for either power gating or leakage reuse is an optimization problem.

3.1 Longest Idle Time - Simultaneous Leakage Reuse and Power Gating Algorithm

A power management algorithm, *longest idle time - simultaneous leakage reuse and power gating (LIT-LRPG)*, is developed that dynamically assigns each idle donor core in a MPSoC for either power gating or leakage current reuse. The pseudo code describing *LIT-LRPG* is provided as Algorithm 2. The *LIT-LRPG* dynamically applies a core idleness ranking to each core, which is similar to Algorithm 1.

A two-bit core tracker $core_t[m]$ is utilized in Algorithm 2 to represent four core activity scenarios: 1) 00 = $D_{CORE,m}$ is active, 2) 01 = $D_{CORE,m}$ is idle (not currently used for LR or PG), 3) 10 = $D_{CORE,m}$ is in use with LR, and 4) 11 = $D_{CORE,m}$ is in use with PG. The priority variable $\delta[m]$ defines which technique (LR or PG) provides greater energy savings for an idle donor core and, therefore, assigns the core to either power gating or leakage current reuse based on the value of the variable. For a $\delta[m]$ of 1, the idle core m is assigned for leakage current reuse, while the remaining idle cores, starting from the core with the second longest idle time, are assigned for power gating. For a $\delta[m]$ of 0, the idle core m is assigned for power gating, while the core with the second longest idle time is assigned for leakage current reuse. All remaining idle cores are assigned for power gating beginning with the core with the third longest idle time.

Algorithm 2 Simultaneous implementation of LR and PG

```
Input: $m$, $\delta[m]$, $N_{BE,PG}$, $N_{BE,LR}$, $Pcycles_{task}$, two-bits core tracker for each of $m$ cores
$core_t[m]$, which keeps track of each core for possible use in LR and PG mode
Output: core assignment for leakage reuse ($LR_{cores}$) and power gating ($PG_{cores}$)
Procedure: Assign idle cores to both LR and PG
 1: ▹ The following section (lines 2 to 36) replaces lines 8 to 15 in Algorithm 1
 2: for $core$ $1, 2, \ldots, m$ do
 3:         max    $X[m] \cdot E_{saved,LR}[m]\;\; Y[m] \cdot E_{saved,PG}[m]$
            s.t.   $LR_{wake-up}[m] \cdot X[m]\;\; PG_{wake-up}[m] \cdot Y[m] \leq L_{max}[m]$
                   $E_{loss,LR}[m] \cdot X[m]\;\; E_{loss,PG}[m] \cdot Y[m] \leq E_{loss,max}[m]$
                   $X[m], Y[m] \geq 0$
 4:     if $X[m] > Y[m]$ then
 5:         $\delta[m] \Leftarrow 1$
 6:     else
 7:         $\delta[m] \Leftarrow 0$
 8:     end if
 9:     if $core_t[m] == 01$ then ▹ Core $m$ is idle and not used for either LR or PG
10:         if $\delta[m] == 1$ then ▹ LR is selected
11:             if $i \neq (Task_{num}$ - 1) then
12:                 $LR_{cores\_index}[i] \Leftarrow max(Core_{1,ranking}, Core_{2,ranking}, Core_{3,ranking}, \ldots$
                      $Core_{m,ranking})$ ▹ Assigning the core with longest idle time for LR
13:                 $LR_{cores}[i] \Leftarrow Order_{core}\left[LR_{cores\_index}[i]\right]$
14:                 $PG_{second,max} \Leftarrow$ sort and assign core index with second longest idle time
15:                 $PG_{cores}[i] \Leftarrow Order_{core}\left[PG_{second,max}\right]$
16:                 $core_t[m] \Leftarrow 10$
17:                 $core_t\left[PG_{second,max}\right] \Leftarrow 11$
18:             else if $i == (Task_{num}$ - 1) then
19:                 $LR_{cores}[i] \Leftarrow LR_{cores}[0]$ ▹ Tracking the core with longest idle time from the
                      execution of last application and assigning for current state
20:                 $PG_{cores}[i] \Leftarrow LR_{cores}[1]$ ▹ Tracking the core with second longest idle time
                      from the execution of last application and assigning for current state
21:             end if
22:         else if $\delta[m] == 0$ then ▹ PG is selected
23:             if $i \neq (Task_{num}$ - 1) then
24:                 $PG_{cores\_index}[i] \Leftarrow max(Core_{1,ranking}, Core_{2,ranking}, Core_{3,ranking}, \ldots$
                      $Core_{m,ranking})$ ▹ Assigning the core with longest idle time for power gating
25:                 $PG_{cores}[i] \Leftarrow Order_{core}\left[PG_{cores\_index}[i]\right]$
26:                 $LR_{second,max} \Leftarrow$ sort and assign core index with second longest idle time
27:                 $LR_{cores}[i] \Leftarrow Order_{core}\left[LR_{second,max}\right]$
28:                 $core_t[m] \Leftarrow 11$
29:                 $core_t\left[PG_{second,max}\right] \Leftarrow 10$
30:             else if $i == (Task_{num}$ - 1) then
31:                 $PG_{cores}[i] \Leftarrow PG_{cores}[0]$ ▹ Tracking the core with longest idle time from the
                      execution of last application and assigning for current state
32:                 $LR_{cores}[i] \Leftarrow PG_{cores}[1]$ ▹ Tracking the core with second longest idle time
                      from the execution of last application and assigning for current state
33:             end if
34:         end if
35:     end if
36: end for
```

Assigning $\delta[m]$ at design time such that a fixed priority is set for LR and PG is possible, where one is assigned with a core with the longest idle time and another with a core with the second longest idle time. However, a dynamic assignment of $\delta[m]$ is implemented through linear programming as described by lines 3 through 8 in Algorithm 2. The variables X and Y define the impact of, respectively, LR and PG on the overall energy efficiency of the circuits in a MPSoC, where a higher value indicates greater energy efficiency and results in the assignment of δ to logic high. The variables $LR_{wake-up}$ and $PG_{wake-up}$ are the wake up latencies normalized to the cycle

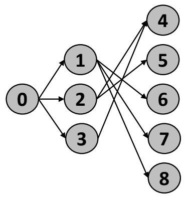
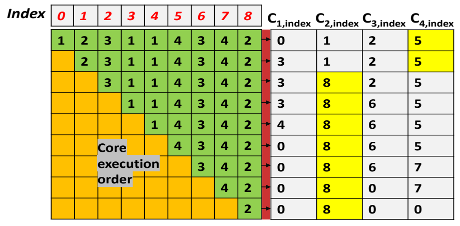


Figure 2: DAG with nine tasks representing the executing application used to characterize *LIT-LR* and *LIT-LRPG* algorithms.

time for, respectively, the leakage reuse and power gating techniques. The energy loss normalized to the total energy consumed by the donor core during the idle state is given by $E_{loss,LR}$ and $E_{loss,PG}$ for, respectively, the leakage reuse and power gating techniques. The energy loss is a function of the energy consumption of an idle donor core $E_{donor,idle}$, the technology dependent fitting parameter η, the energy consumed by the LR LCBs $E_{LCB,LR}$, and the energy consumed by the PG LCBs $E_{LCB,PG}$ and is given by $E_{loss,LR} = \frac{\eta \cdot E_{LCB,LR}}{E_{donor,idle}}$ and $E_{loss,PG} = \frac{\eta \cdot E_{LCB,PG}}{E_{donor,idle}}$ for, respectively, the LR and PG techniques. The savings in energy normalized to the total energy consumed for the leakage reuse and power gating techniques is given by $E_{saved,LR} = \frac{E_{donor,idle} - E_{LCB,LR} + E_{receiver,LR}}{E_{total}}$ and $E_{saved,PG} = \frac{E_{donor,idle} - E_{LCB,PG}}{E_{total}}$, respectively. In addition, L_{max} describes the maximum allowed latency normalized to the cycle time, and $E_{loss,max} = \frac{E_{donor,idle}}{E_{total}}$ defines the maximum allowed energy loss normalized to the total energy.



Index	0	1	2	3	4	5	6	7	8	$C_{1,index}$	$C_{2,index}$	$C_{3,index}$	$C_{4,index}$
	1	2	3	1	1	4	3	4	2	0	1	2	5
		2	3	1	1	4	3	4	2	3	1	2	5
			3	1	1	4	3	4	2	3	8	2	5
				1	1	4	3	4	2	3	8	6	5
					1	4	3	4	2	4	8	6	5
						4	3	4	2	0	8	6	5
							3	4	2	0	8	6	7
								4	2	0	8	0	7
									2	0	8	0	0

Core execution order

Figure 3: Identification of donor cores with longest idle time that are assigned for leakage current reuse using the *LIT-LR* algorithm.

4 SIMULATION FRAMEWORK

The activation and deactivation of a power gating mode results in voltage transients on the power distribution network (PDN) due to the switching of the control transistors implementing the power gating [5, 17]. The proposed technique that reuses leakage current from idle cores results in similar switching voltage transients. Therefore, the voltage noise due to the implementation of the leakage reuse and power gating techniques is characterized while accounting for the impedance of the power network. In this work, an optimized PDN with off-chip and on-chip impedance parameters is implemented that allows for a more accurate analysis of the transient noise.

Both the *LIT-LR* and *LIT-LRPG* algorithms are evaluated through SPICE simulation using a 45 nm CMOS process. A MPSoC platform with five cores is utilized, where four homogeneous cores are implemented as donors and one core is implemented as a receiver. The donor cores operate at a supply voltage of 1.2 V with a 1 GHz clock frequency, while a sub-threshold (Sub-V$_t$) supply voltage of 340 mV is generated for the receiver core from the reused leakage current of one of the four donor cores. The receiver core operates at 100 MHz. Note that the supply voltage of the receiver core is adjustable based on the ratio of the areas between the donor and receiver cores as mentioned in Section 2.1. The ISCAS89 benchmark circuits and boolean logic circuits are implemented within each of the four donor cores, while the receiver core is implemented as either the ISCAS89 s27 benchmark circuit or a 32-bit ARM Cortex M0 core.

An application with nine periodic tasks is used to characterize non-preemptive scheduling of donor cores, where the task graph is as shown in Fig. 2. The execution order of four donor cores is shown in Fig 3. During the execution of a task on a given donor core, the remaining donor cores are idle and, therefore, available for *LIT-LR* and *LIT-LRPG* scheduling. The longest idle period for each core when executing the nine tasks is determined through the *LIT-LR* algorithm. For example, Task 0 is executed on core 1 as shown in Fig 3, while core 4 provides the longest idle time before starting execution of a task and, therefore, is assigned with the largest index value ($C_{4,index} = 5$). In this case, core 4 is assigned for leakage current reuse when executing Task 0 on core 1. Similarly, core 2 provides the longest idle time when executing third task on core 3 and is, therefore, assigned for leakage current reuse. Note that the index number and task number are not the same.

An accurate estimation of the number of cycles corresponding to the break-event points ($N_{BE,LR}$ and $N_{BE,PG}$) requires a separate analysis. In this paper, $Pcycles_{task}$ is assumed as larger than both $N_{BE,LR}$ and $N_{BE,PG}$ for the entire task graph since only one receiver core implements leakage current reuse and four homogeneous donor cores are used for both LR and PG. To the best of our knowledge, this is the first work that addresses the scheduling of idle cores for leakage current reuse. Therefore, the proposed *LIT-LR* algorithm is compared against random assignment of available idle donor cores for leakage current reuse. Based on the idle core assignment performed by the *LIT-LR* and *LIT-LRPG* algorithms, control bits are set and provided to the LCB for SPICE simulation. For the *LIT-LR* algorithm, a one bit control signal Φ is generated and supplied, while a two-bit control signal is generated and supplied to the LCB after executing the *LIT-LRPG* algorithm. Note that the LCB circuit is similar to the control circuit used in [12] and [13], and the same number of transistors are used in each LCB for both the *LIT-LR* and *LIT-LRPG* algorithms.

Table 1: Off-chip and on-chip parameters of the simulated power delivery network [13].

Resistance	Value (Ω)	Inductance	Value (H)	Capacitance	Value (F)
R_{vrm}	100u	L_{vrm}	5n	C_{bulk}	132u
R_{pcb}	485u	L_{bulk}	510p	C_{pcb}	10u
R_{bulk}	4.365m	L_{pcb}	48p	C_{pkg}	440n
R_{ppcb}	10m	L_{pkg}	19p	C_{odc}	1n
R_{pkg}	123u	L_{ppkg}	104p	$C_{odc,receiver}$	50p
R_{ppkg}	15.72m	L_{bump}	38p		
R_{bump}	400u	L_{grid}	5.6f		
R_{grid}	50m				

Optimized PDN with Off-Chip and On-Chip Impedance Model: The four donor cores are supplied current through an optimized PDN that includes both off-chip impedance components and on-chip distributed power and ground networks as shown in Fig. 4. The off-chip and on-chip impedance parameters are listed in Table 1 [13]. A 4×5 on-chip grid is used to improve the simulation accuracy without significantly increasing the computational cost. The output of the off-chip voltage regulator module (VRM) is maintained at 1.2 V to emulate the closed-loop response of a VRM.

On-chip decoupling capacitors (C_{odc}) are placed at each load point to compensate for any voltage ripples, where a total of 4 nF of decoupling capacitance is implemented for the four donor cores. In addition, an on-chip decoupling capacitor $C_{odc,receiver}$ of 50 pF is implemented on the supply node of the receiver core to minimize any voltage ripple on the virtual ground node. The frequency response of the PDN is evaluated. A maximum impedance of 320 mΩ occurs at 700 MHz, which is much lower than the target impedance of the PDN given by $\frac{V_{DD} \times \%ripple}{I_{max} - I_{min}}$. The PDN is optimized to maintain the voltage ripple within 10% of the target voltage on both the VDD and GND networks.

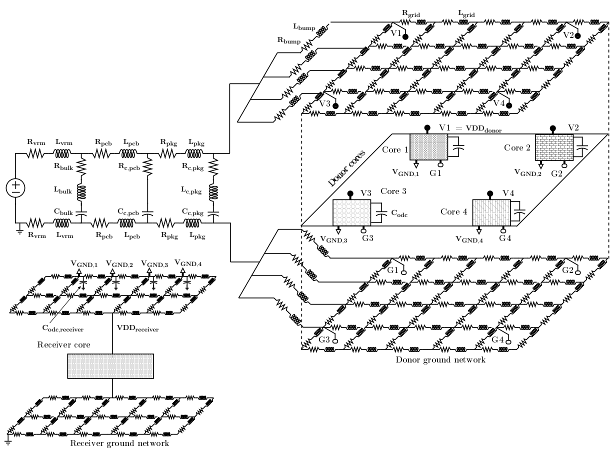


Figure 4: Optimized PDN modeled with off-chip and on-chip impedance to evaluate the *LIT-LR* and *LIT-LRPG* algorithms.

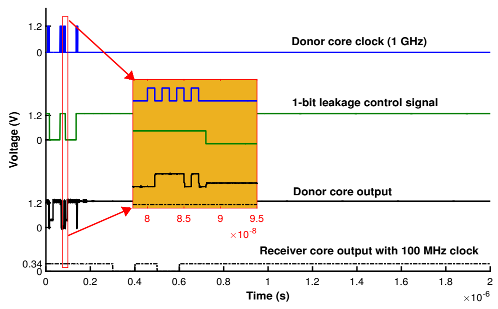


Figure 5: SPICE simulation results characterizing the activity of both the donor and receiver cores when executing the leakage reuse technique when applying the *LIT-LR* algorithm.

5 RESULTS AND DISCUSSION

The *LIT-LR* algorithm is evaluated through SPICE simulation using the circuit shown in Fig. 4 for the `ss, tt, and ff process corners`. The voltage transients due to the leakage reuse technique are characterized for three different sizes of the switches within each LCB, where switches with 7.5%, 15%, and 30% of the area of a donor core are used.

Table 2: Execution order of four *donor* cores and the corresponding assignments by the *LIT-LR* and *LIT-LRPG* algorithms for leakage current reuse and power gating.

Execution order of donor cores		1	2	3	1	1	4	3	4	2
Idle core assignment	Leakage current recycling: LIT-LCR	4	4	2	2	2	2	2	2	4
	Leakage current recycling: Random	2	1	4	2	3	1	2	3	1
	Power gating (LIT-LCRPG)	3 2	1 3	4 1	3 4	3 4	3 1	4 1	1 3	3 1

The execution order generated by the HEFT algorithm for the DAG shown in Fig.2, the idle donor core assignments produced by the *LIT-LR* algorithm, and a random allocation of cores is listed in Table 2. The control bits generated by both *LIT-LR* and a random assignment are provided to the LCB for SPICE simulation. The transient behavior of the leakage current reuse technique when executing the *LIT-LR* algorithm is shown in Fig. 5, where the area of the LCB is 7.5% that of a donor core. The outputs of a donor and a receiver core are plotted in Fig. 5, which operate at a clock frequency of, respectively, 1 GHz and 100 MHz. Despite the reuse of leakage current, a full voltage swing is maintained at every clock cycle at the outputs of all four donor cores (the output of only one donor core is shown). In addition, a steady voltage of 340 mV is maintained at the output of the receiver core. Additional characterization of the maximum and minimum voltage noise on the supply node of the receiver core (the *VGND* node) is performed while considering process variation and for three different sized LCBs. The results of the characterization are listed in Table 3. Even with large MOS switches (up to 30% of the donor core) in each LCB and for the extreme process corners (`ss` and `ff`), the maximum and minimum supply voltage of the receiver core remains within ±3% of 340 mV. Therefore, the proposed leakage current reuse technique and scheduling algorithm provide a stable supply voltage for the receiver core with the use of only on-chip decoupling capacitors.

Table 3: Voltage generated on the receiver core from reused charge from idle donor cores for different switch sizes and process corners.

Switch size (%)	7.5			15			30		
Corner	*SS*	*TT*	*FF*	*SS*	*TT*	*FF*	*SS*	*TT*	*FF*
Max. $Sub-V_t$ **VDD (mV)**	336.5	340.6	345.4	337.2	341.2	345.9	338.6	342.4	346.9
Min. $Sub-V_t$ **VDD (mV)**	336.2	340.2	344.8	336.9	340.8	345.4	338.2	342	346.4

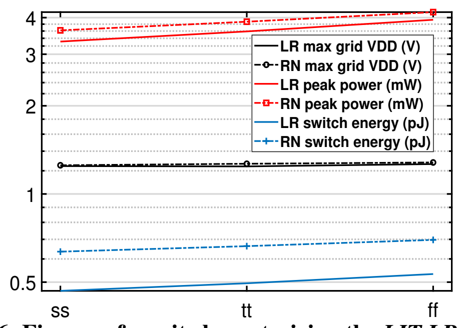


Figure 6: Figures of merit characterizing the *LIT-LR* algorithm (solid lines) and random core assignment (dotted lines).

The maximum noise on the supply voltage *VDD*, the peak power consumption of the MPSoC system (the sum of the power consumed by the donor cores, receiver core, and LCBs), and the total energy consumed by the LCBs due to the execution of the *LIT-LR* assignment and the random core assignment are characterized for the `ss`, `tt`, and `ff` process corners with the results shown in Fig 6. The assignment of idle donor cores using the *LIT-LR* algorithm reduces the energy consumed by the switches by 27%, 25%, and 24% in, respectively, the `ss`, `tt`, and `ff` corners as compared to the random assignment of cores. The peak power consumption when assigning cores with *LIT-LR* is reduced by 8.4%, 7.4%, and 5.8% as compared to the random assignment of cores in, respectively, the `ss`, `tt`, and `ff` corners. In addition, the maximum voltage noise on the on-chip distributed PDN is reduced by 0.64%, 2.1%, and 1.3% when *LIT-LR* assignment is performed in, respectively, the `ss`, `tt`, and `ff` corners as compared to random assignment.

Similar to the *LIT-LR* algorithm, the control bits generated from execution of the *LIT-LRPG* algorithm are supplied to the LCBs shown in Fig. 1 to evaluate the simultaneous implementation of LR and PG. SPICE simulation is performed at the `tt` corner and with a LCB area of 7.5% that of a donor core. However, for the simultaneous implementation of both techniques, two-bit control signals are passed to each LCBs: one bit to S_{VG} and another to S_G.

The simulated results obtained from a 45 nm CMOS technology are used to solve the optimization problem described in line 3 of Algorithm 2. The $LR_{wake-up}$ and $PG_{wake-up}$ parameters, which are extracted from simulation, are set to, respectively, 1.1 and 1.2 for a 1 GHz clock frequency. The $E_{donor,idle}$, $E_{LCB,LR}$, $E_{LCB,PG}$, and E_{total}

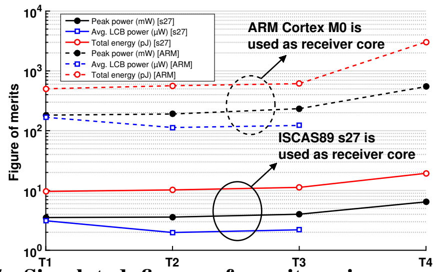


Figure 7: Simulated figure of merits using proposed Algorithm 2, where T1, T2, T3, T4 represents, respectively, simultaneous implementation of LR and PG, LR only, PG only, and baseline without LR and PG.

parameters are determined as, respectively, 4.75 pJ, 0.16 pJ, 0.63 pJ, and 29.2 pJ. Consequently, an optimum solution is found with a larger value of X by setting $E_{loss,LR}$, $E_{loss,PG}$, $E_{saved,LR}$, $E_{saved,PG}$, L_{max}, $E_{loss,max}$, and η to, respectively, 0.33, 1.33, 0.162, 0.14, 5, and 0.17, and 10. As homogeneous cores are used, the value of $\delta\,[m]$ is the same for all donor cores. Therefore, in this case, the idle donor core with the longest idle period is always assigned for leakage current reuse.

The peak power, average power consumption per cycle, and total energy consumption are characterized for four power management scenarios: T1) simultaneous implementation of leakage current reuse and power gating, T2) power gating only, T3) leakage current reuse only, and T4) without either leakage current reuse or power gating (baseline). The simulation is performed with two sizes of receiver cores: one that contains only an s27 benchmark circuit (3.36 μm^2) and another that consists of a 32-bit ARM Cortex M0 core (98930 μm^2). Both s27 and the synthesized ARM core operate with a supply voltage of 340 mV. The current trace for 10% activity of the ARM core is applied to the PDN shown in Fig. 4 for SPICE simulation. The size of each donor core is increased by 1750× (equivalent to 1750 s27 blocks) to provide sufficient leakage current to the ARM core operating at 340 mV.

The simulation results for the four scenarios are shown in Fig. 7. The simultaneous implementation of LR and PG reduces the total energy consumption of the donor cores by 50.2% (75.5%), 14.4% (18%), and 5.7% (11.2%) for receiver cores consisting of s27 (ARM Cortex M0) as compared to, respectively, the baseline with no PG or LR, power gating only, and leakage current reuse only. The use of leakage current reuse also reduces the average power consumption per cycle of the LCB by 9.9% (7.6%) as compared to power gating when applied to receiver core consisting of s27 (ARM core). In addition, the simultaneous implementation of PG and LR on s27 (ARM core) reduces the peak power consumption by 45.2% (66.7%), 11.9% (22%), and 1.62% (4.2%) as compared to, respectively, the baseline, PG only, and LR only. Although the simultaneous implementation of LR and PG results in a greater power consumption by the LCB than power gating or leakage current reuse implemented separately, the energy consumption and peak power consumption of the overall MPSoC is reduced.

6 CONCLUSIONS

In this paper, circuit and algorithmic techniques for dynamic idle core management are proposed to reduce the leakage current and improve the overall system energy efficiency of a MPSoC platform. The leakage current of idle donor cores is reused to generate a supply voltage of 340 mV with less than ±3% variation in voltage for a receiver core operating in sub-threshold. A novel longest idle time based algorithm *LIT-LR* is proposed to dynamically assign the idle cores for leakage current reuse. The execution of the *LIT-LR* algorithm on a MPSoC with five cores resulted in a reduction of the energy consumption by the LCBs (switches) and the peak power consumption by, respectively, 25% and 7.4% as compared to a random assignment of cores for LR. In addition, a novel *LIT-LRPG* algorithm to simultaneously implement leakage current reuse and power gating is proposed, where the overall energy efficiency of a MPSoC platform is maximized when both leakage current reuse and power gating are concurrently applied. The analysis of the *LIT-LRPG* algorithm through SPICE simulation indicates that the total energy consumption is reduced by 50.2%, 14.4%, and 5.7% as compared to, respectively, the baseline scenario that includes neither leakage reuse or power gating, only power gating, and only leakage current reuse. In addition, leakage current reuse reduces the power consumption of the LCBs by 9.9% as compared to the power gating technique. Therefore, the proposed dynamic idle core management techniques are suitable for both individual implementation and with any existing power gating techniques to improve the overall system energy-efficiency.

REFERENCES

[1] M. Arora, S. Manne, I. Paul, N. Jayasena, and D. M. Tullsen, "Understanding Idle Behavior and Power Gating Mechanisms in the Context of Modern Benchmarks on CPU-GPU Integrated Systems," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 366–377, March 2015.

[2] A. Esmaili, M. Nazemi, and M. Pedram, "Modeling Processor Idle Times in MPSoC Platforms to Enable Integrated DPM, DVFS, and Task Scheduling Subject to a Hard Deadline," *Proceedings of the IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 532–537, January 2019.

[3] R. Ye and Q. Xu, "Learning-based Power Management for Multicore Processors via Idle Period Manipulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 33, No. 7, pp. 1043–1055, July 2014.

[4] Q. Diao and J. Song, "Prediction of CPU Idle-Busy Activity Pattern," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 27–36, October 2008.

[5] Z. Wang, X. Wang, J. Xu, H. Li, R. K. Maeda, Z. Wang, P. Yang, L. H. Duong, and Z. Wang, "An Adaptive Process-Variation-Aware Technique for Power-Gating-Induced Power/Ground Noise Mitigation in MPSoC," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 24, No. 12, pp. 3373–3386, December 2016.

[6] S. Kim, S. V. Kosonocky, D. R. Knebel, and K. Stawiasz, "Experimental Measurement of a Novel Power Gating Structure with Intermediate Power Saving Mode," *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 20–25, August 2004.

[7] K. Agarwal, K. Nowka, H. Deogun, and D. Sylvester, "Power Gating with Multiple Sleep Modes," *Proceedings of the IEEE/ACM International Symposium on Quality Electronic Design*, pp. 633–637, March 2006.

[8] H. Cherupalli, H. Duwe, W. Ye, R. Kumar, and J. Sartori, "Enabling Effective Module-oblivious Power Gating for Embedded Processors," *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 157–168, February 2017.

[9] N. Nasirian, R. Soosahabi, and M. A. Bayoumi, "Probabilistic Analysis of Power-Gating in Network-on-Chip Routers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 66, No. 2, pp. 242–246, February 2018.

[10] D. Zoni, L. Colombo, and W. Fornaciari, "Darkcache: Energy-performance Optimization of Tiled Multi-cores by Adaptively Power-gating LLC Banks," *ACM Transactions on Architecture and Code Optimization*, Vol. 15, No. 2, pp. 21, June 2018.

[11] H. Singh, K. Agarwal, D. Sylvester, and K. J. Nowka, "Enhanced Leakage Reduction Techniques Using Intermediate Strength Power Gating," *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 15, No. 11, pp. 1215–1224, November 2007.

[12] M. S. Hossain and I. Savidis, "Reusing Leakage Current for Improved Energy Efficiency of Multi-Voltage Systems," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1–5, May 2019.

[13] M. S. Hossain and I. Savidis, "Recycling of Unused Leakage Current for Energy Efficient Multi-voltage Systems," *Microelectronics Journal*, Vol. 101, No. 7, pp. 1–16, July 2020.

[14] K. Blutman, A. Kapoor, J. G. Martinez, H. Fatemi, and J. P. Gyvez, "Lower Power by Voltage Stacking: A Fine-grained System Design Approach," *Proceedings of the ACM/IEEE Design Automation Conference*, pp. 78:1–78:5, June 2016.

[15] T. Nakada, H. Yanagihashi, H. Nakamura, K. Imai, H. Ueki, T. Tsuchiya, and M. Hayashikoshi, "Energy-aware Task Scheduling for Near Real-time Periodic Tasks on Heterogeneous Multicore Processors," *IEEE International Conference on Very Large Scale Integration*, pp. 1–6, December 2017.

[16] H. Topcuoglu, S. Hariri, and M. Wu, "Performance-effective and Low-complexity Task Scheduling for Heterogeneous Computing," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 3, pp. 260–274, August 2002.

[17] H. Jiang and M. Marek-Sadowska, "Power Gating Scheduling for Power/Ground Noise Reduction," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 980–985, June 2008.