2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

Scheduling of Dual Supercapacitor for Longer Battery Lifetime in Systems with Power Gating

Sumanta Pyne
Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Sundargarh, Odisha - 769008, India
Email: pynes@nitrkl.ac.in

***Abstract*—The wake-up of power gating (*PG*) components leads to flow of in-rush current which quickly discharges the battery. An arrangement of instruction controlled hybrid battery-supercapacitor can enable longer battery life in systems with instruction controlled *PG*. The present work extends a battery-single supercapacitor system (*B-SC*) model to its equivalent battery-dual supercapacitor system (*B-2SC*) and shows that *B-2SC* can achieve higher longevity than its equivalent *B-SC*. Two instructions – disconnect battery (*db*) and connect battery (*cb*) have been introduced along with architectural support for *B-2SC*. The instruction *db* disconnects the battery from the *PG* components during wake-up. It also disconnects either one or both of the supercapacitors from the battery. Hence simultaneously either both supercapacitors can discharge or one can discharge while the other can be charged. Disconnecting the battery during wake-up minimizes rate capacity effect (C-rate) for longer battery life. The instruction *cb* connects the battery to the *PG* components and the supercapacitors. The *db-cb* scheduler inserts a *db* before an instruction which causes to wake-up of *PG* components. It places a *cb* in such a way that it executes after wake-up of all the desired the *PG* components. The efficacy of the proposed method is evaluated on MiBench and MediaBench benchmark programs. *B-SC* and *B-2SC* reduce C-rate by an average of 14.25% and 21.87%, respectively with corresponding average performance loss of 6.87% and 9.25%.**

I. Introduction

The emergence of deep submicron process technology with the decrease in dimensions of the transistor has increased the transistor count and speed of operation at the cost of greater device leakage currents. Power gating (*PG*) is a technique used to reduce power consumption of VLSI chips, by shutting off the blocks that are not in use, thus reducing stand-by or leakage power. Shutting down of the blocks can be initiated either by software or by hardware. Many modern processors are equipped with *PG* instructions to switch-off and switch-on different blocks to sleep and active modes, respectively. This allows the compiler and operating system to reduce runtime leakage power.

When a power gated block is switched on from sleep mode to active mode it draws a huge amount of in-rush current due to simultaneous charging of its internal capacitors. In-rush current is several times higher than the actual current required by the block to function in active mode. The flow of in-rush current may cause permanent damage to the circuit and also lead to higher power consumption. It can reduce the battery life for battery-operated systems due to rise in load current.

Supercapacitors are suitable for applications requiring a large amount of energy to be stored and delivered in bursts repeatedly. Their rapid charging and discharging property make them an ideal to systems requiring high peak currents [1]. The present work improves the battery-single supercapacitor (*B-SC*) scheduling proposed in [2] to an equivalent battery-dual supercapacitor (*B-SC*) scheduling. The *B-2SC* scheduling for systems with *PG* is based on hardware/software co-design. This is done by insertion of disconnect battery (*db*) and connect battery (*cb*) instructions to a program with *PG* instructions. The battery is disconnected from the *PG* components during their activation allowing the in-rush current to be drawn from the supercapacitors. This helps in reduction of rate capacity effect (C-rate) resulting increase in battery lifetime. The battery is connected to the *PG* components and supercapacitors after the desired *PG* components are activated. This allows the battery to provide current required for normal operation of *PG* components and recharging of supercapacitors. This approach can enable a compiler to generate target code for higher battery lifetime in battery driven systems with *PG*.

The existing research and development works on increment of battery lifetime are briefly discussed in Sec. II. The proposed technique is explained in Sec. III. Section IV covers explaination of the experimental setup with analysis of the results. Finally, Sec. V concludes the present work with future directions.

II. Related Works

The earliest works on enhancement of battery lifetime were based on recovery effect. The battery scheduling schemes proposed in [3], [4], [5] enhance battery lifetime in multi-battery systems by exploiting battery recovery effect. The Chiasserini-Rao [6], stochastic KiBaM [7] and stochastic network based continuous-time Markovian decision models [9] were developed considering recovery effect. Dougal et al. in [1] provided a simplified analytical model to describe the performance of a *B-SC* power source under pulsed load conditions. They showed that peak power can be greatly enhanced with considerably reduced internal losses and extended discharge life of the battery. Shin et al. in [8] presented a new *B-SC* system having a constant-current charger which isolates the battery from supercapacitor to improve the end-to-end efficiency for

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00045

energy from the battery to the load while accounting for the C-rate of Li-ion batteries and the conversion efficiencies of the converters. In [11] Narayanaswamy et al. proved the non-existence of charge recovery effect in batteries through their detailed experimental evaluations. They identified that the C-rate as the dominant electrochemical phenomenon which can be minimized to obtain higher energy output from the battery through both hardware and software based power management approaches. They used a *B-SC* arrangement in [8] to minimize battery C-rate in *WSN*s. They showed that the supercapacitor handles the higher peak power experienced during an intermittent discharge and is efficiently charged at a lower continuous discharge rate from the battery using the DC-DC converter. Hence the *B-SC* model is a suitable candidate for longer battery life in *PG* systems.

Scheduling of hybrid battery-supercapacitor considering a battery-single supercapacitor (*B-SC*) model for systems with instruction controlled *PG* has been introduced in [2]. The instruction *db* disconnects the battery from the *PG* components if the charge in the supercapacitor greater than or equal to the charge required by wake-up of *PG* components. The instruction *cb* connects the battery to the *PG* components and the supercapacitors after the wake-up of the *PG* components. The proposed *db-cb* scheduling algorithm inserts a *db* before the wake-up of *PG* components and places a *cb* such that it executes after all desired *PG* components are active.

III. Present Work

An arrangement for instruction controlled *PG* is shown in Fig. 1. It has n *PG* components $C_0, C_1, \cdots, C_{n-1}$. *PG* is done with the help of the header p-MOS transistors having higher threshold voltage (V_{th}). The header switches are controlled by an n-bit power gating control register (PGCR) placed in the power gating controller (PGC). The bits $0, 1, \cdots, n-1$ are the *PG* bits of $C_0, C_1, \cdots, C_{n-1}$, respectively. If any of these bits $\alpha \in \{0, 1, \cdots, n-1\}$ is '0', then the component C_α is in active mode, otherwise C_α is in sleep mode. Let there be two *PG* instructions *switch_off* and *switch_on*. To put C_α in sleep (or power gated) mode the instruction *switch_off*(C_α) is used to set the value of α^{th} bit of PGCR. C_α in sleep mode can be put to active mode with the help of the instruction *switch_on*(C_α) which resets the value of α^{th} bit of PGCR. Hence, a program can use this *PG* facility. The high level *PG* instructions *switch_off* and *switch_on* are designed to support *PG* in high level languages. They are replaced by an assembly language level instruction ***pg pgcr_bit_vector*** which has been added to the instruction set, where *pgcr_bit_vector* is a 32-bit vector representing the *PG* bits of PGCR. Its size is five bytes. It sets/resets corresponding *PG* bits of PGCR to switch OFF/ON the *PG* components in three clock cycles - one cycle in each of instruction fetch (*IF*), instruction decode (*ID*) and execution (*EX*) stages of the instruction pipeline.

When C_α in sleep mode is switched on using *switch_on*(C_α) it draws in-rush current I_α for a period of w_α cycles where w_α is the wake-up time (T_w) of C_α. The problem of intolerable

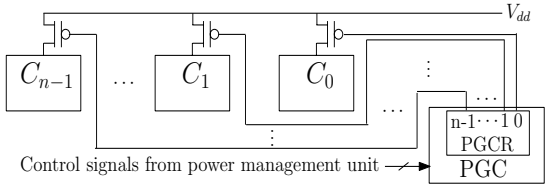
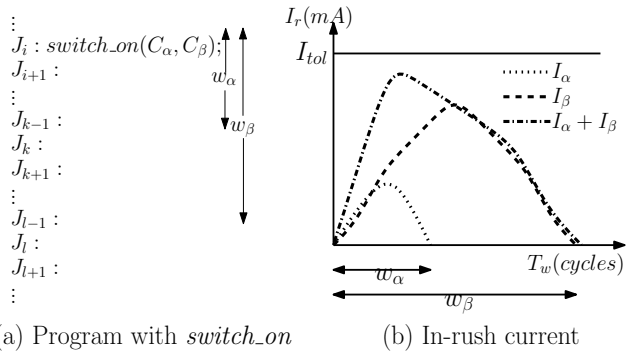


Fig. 1. An arrangement for instruction controlled *PG* system



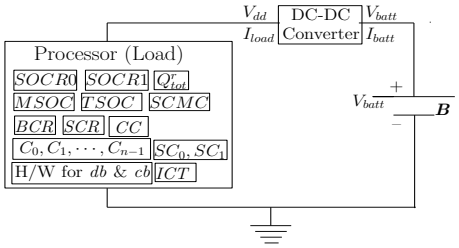
(a) Program with *switch_on* (b) In-rush current

Fig. 2. In-rush current due to overlapped wake-up

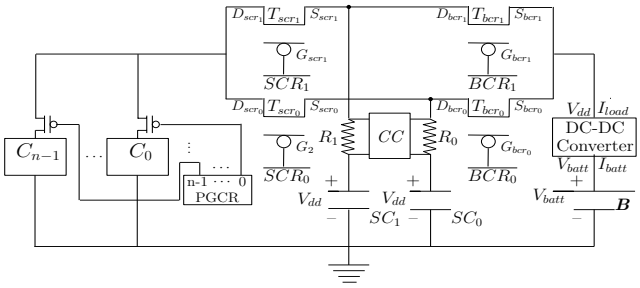
in-rush current may arise during wake-up of multiple components during an overlapped time interval. A program with *switch_on* instructions may cause overlapped wake-up of *PG* components. Let there be an assembly language program with m instructions where i^{th} instruction J_i is *switch_on*(C_α, C_β) as shown in Fig. 2(a) where $i, k, l \in \{0, 1, \cdots, m-1\}, i < k < l, \alpha \neq \beta$ and $\beta \in \{0, 1, \cdots, n-1\}$. Figure 2(b) shows in-rush current (I_r) in milliampere (mA) for overlapped wake-up of two *PG* components C_α and C_β. w_β and I_β are the wake-up time and in-rush current for C_β, respectively. The resultant in-rush current is $I_\alpha + I_\beta$. I_{tol} is the maximum tolerable in-rush current for a given system. Simultaneous overlapped wake-up of several components can lead to higher flow of in-rush current resulting higher peak power dissipation and reduction of chip reliability. Hence, it is better to avoid overlapped wake-up. It is considered that the magnitudes of all possible overlapped in-rush current are within the limit of I_{tol}.

A. Instruction controlled battery-dual supercapacitor system

A battery powered processor with *n PG* components having hardware support for battery control and *PG* instructions for *B-2SC* are shown in Fig. 3(a). The detailed circuitry for instruction controlled *B-2SC* system is shown in Fig. 3(b). Battery *B* and supercapacitors $SC_\sigma \forall \sigma \in \{0, 1\}$ are in parallel configuration. The processor (load) is powered by battery *B*. The *PG* components are connected to each other in a parallel configuration. As a whole the *PG* components are connected to the *B-2SC* system in series. *B* is disconnected from the *PG* components during their wake-up. The rest of the processor other that the *PG* components remains connected to *B*. SC_σ is discharged to activate the *PG* components. After activation *B* is connected to the *PG* components and SC_σ. For normal operation the *PG* components draw current from *B*. For recharging the SC_σ draws current from *B*. A DC-DC converter is placed in series with *B*. It converts the battery



(a) A processor with PG and *B-2SC*



(b) PG components with instruction controlled *B-2SC*

Fig. 3. Instruction controlled *B-2SC* system

supply voltage V_{batt} and current I_{batt} to processor's operating voltage V_{dd} and load current I_{load}. V_{batt} and V_{dd} are constants while I_{load} varies with processor workload. The capacities of the battery and the supercapacitor in *B-SC* are considered as Q_B and Q_{SC} charge units, respectively [2]. In *B-2SC* the capacities of B and each SC_σ are considered to be Q_B and $\frac{Q_{SC}}{2}$ charge units, respectively where $SC_\sigma \forall \sigma \in \{0, 1\}$ are connected to each other in a parallel configuration.

The processor with *B-2SC* has a 2-bit battery control register (*BCR*) and a 2-bit supercapacitor control register (*SCR*). Bit BCR_σ helps to connect and disconnect *B* from $C_0, C_1, \cdots, C_{n-1}$ and SC_σ where $\sigma \in \{0, 1\}$. This is done by controlling the gate (G_{bcr_σ}) voltage of p-MOS transistor T_{bcr_σ} acting as a load switch, where the source S_{bcr_σ} is connected directly to the input voltage rail, and the drain D_{bcr_σ} is connected to SC_σ and source S_{scr_σ} of the p-MOS transistor T_{scr_σ}. Bit SCR_σ helps to connect and disconnect SC_σ from $C_0, C_1, \cdots, C_{n-1}$ where $\sigma \in \{0, 1\}$. This is done by controlling the gate (G_{scr_σ}) voltage of p-MOS transistor T_{scr_σ} acting as a load switch, where the source S_{scr_σ} is connected to SC_σ and drain D_{bcr_σ} of T_{bcr_σ}, and the drain D_{scr_σ} is directly connected to $C_0, C_1, \cdots, C_{n-1}$. Table I shows the behaviour of *B-2SC* system based on the possible contents of *SCR* and *BCR*.

TABLE I
CONTENT OF *SCR-BCR* WITH STATUS OF SC_0, SC_1 AND *B* IN *B-2SC*

SCR_1	SCR_0	BCR_1	BCR_0	SC_0 gets	SC_1 gets	*B* connected to
$(0)_2$	$(0)_2$	$(0)_2$	$(0)_2$	charged	charged	$SC_0, SC_1, C_0, C_1, \cdots, C_{n-1}$
$(0)_2$	$(0)_2$	$(1)_2$	$(1)_2$	discharged	discharged	none
$(0)_2$	$(1)_2$	$(1)_2$	$(0)_2$	charged	discharged	SC_0
$(1)_2$	$(0)_2$	$(0)_2$	$(1)_2$	discharged	charged	SC_1

Let Q^r_α be the number of charge units required to activate $C_\alpha, \forall \alpha \in \{0, 1, \cdots, n-1\}$. In *B-2SC* SC_σ stores a maximum of $\frac{Q_{SC}}{2}$ charge units where $Q_{SC} = \sum_{\alpha=0}^{n-1} Q^r_\alpha$. The processor in *B-2SC* have $(\lceil log_2 Q_{SC} \rceil - 1)$-bit state of

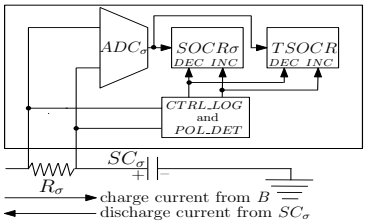


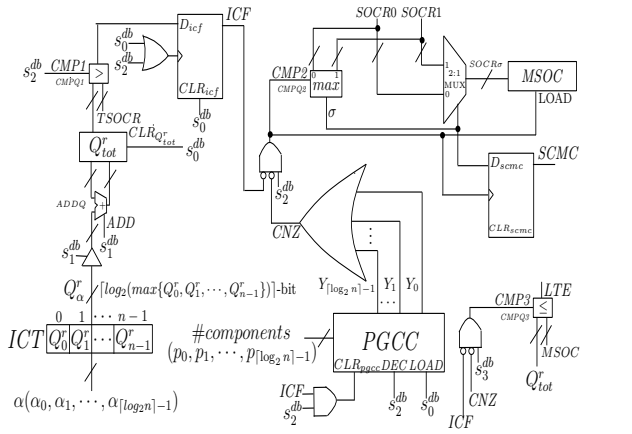
Fig. 4. Coulomb Counter (*CC*) [10] for *B-2SC*

charge register $SOCR\sigma$ for each SC_σ. $SOCR\sigma$ is a real-time up-down counter register that stores the state of charge (*SoC*) information of SC_σ as remaining number of charge units where $SoC \in \left\{0, 1, \cdots, \frac{Q_{SC}}{2}\right\}$. Each SC_σ is associated with a coulomb counter *CC* [10] whose basic operation are shown in Fig. 4. The voltage drop on sense resistor R_σ caused by charge/discharge current to/from SC_σ is converted to number of charge units gained/lost by a $\lceil log_2 Q_{SC} \rceil$-bit analog to digital converter ADC_σ. The control logic *CTRL_LOG* and polarity detector *POL_DET* guides $SOCR\sigma$ to increase/decrease the number of charge units gained/lost. Thus the processor knows the *SoC* of SC_σ in runtime. The processor of *B-2SC* contains another $\lceil log_2 Q_{SC} \rceil$-bit up-down counter register named total state of charge register (*TSOC*) which stores the total *SoC* of the supercapacitors that is $TSOC = \sum_{\sigma=0}^{1} SOCR\sigma$.

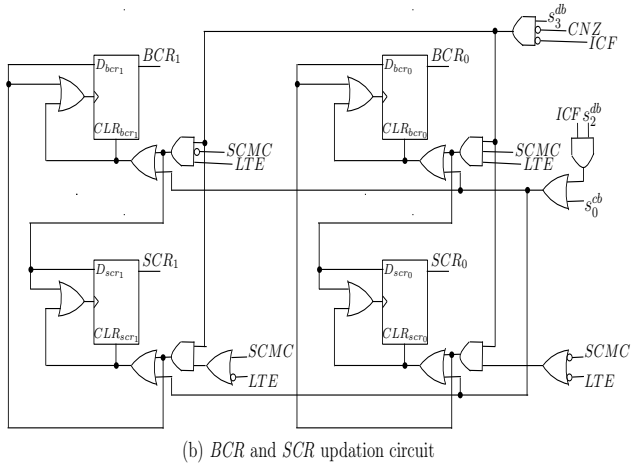
Two assembly level battery control instructions **disconnect battery** (***db***) and **connect battery** (***cb***) for *B-2SC* are included in the instruction set of the processor. The instruction ***db #components,component_list*** disconnects *B* from the *PG* components if the supercapacitor(s) has(have) enough charge required for wake-up of $\#components (\in \{1, 2, \cdots, n\})$ *PG* components in $component\_list$ where $component\_list[c] = \alpha$ for each C_α activated by a *switch_on* and $\forall c | c \in \{0, 1, \cdots, \#components - 1\}$. The instruction ***cb*** connects *B*. ***db*** is a $\#components + 2$ byte instruction. The first two bytes are consumed by its opcode and *#components* while rest are consumed by *component_list*. It consumes a maximum of $3 + 2 \times \#components + 1$ cycles and a minimum of 5 clock cycles. The first two cycles of ***db*** are spent in *IF* and *ID* stages, and rest are in the *EX* stage. The instruction ***cb*** is an one byte instruction as consumed by its opcode. It takes 3 clock cycles, one in each of the *IF*, *ID* and *EX* stages.

B. Supporting Hardware for db and cb

The supporting hardware for ***db*** and ***cb*** of *B-2SC* is shown in Fig. 5. It considers $s^{db}_0, s^{db}_1, s^{db}_2$ and s^{db}_3 as control signal lines for ***db***. s^{cb}_0 is a control signal line for ***cb***. An **in-rush charge table** (*ICT*) is maintained. Each tuple $ICT[\alpha]$ of *ICT* stores Q^r_α the charge required to activate $C_\alpha, \forall \alpha \in \{0, 1, \cdots, n-1\}$. $ICT[\alpha]$ consumes x bits where $x = \lceil \mathbf{log_2}(\mathbf{max}\{Q^r_0, Q^r_1, \cdots, Q^r_{n-1}\}) \rceil$. The $\lceil \mathbf{log_2} n \rceil$-bit **power gating component counter** (*PGCC*) stores the number of *PG* components to be turned on. It is loaded with *#components* field of a *switch_on* instruction through the input lines $p_0, p_1, \cdots, p_{\lceil \mathbf{log_2} n \rceil - 1}$ when its control line



(a) Micro-operations for $s_0^{db}, s_1^{db}, s_2^{db}$ and s_3^{db} (upto generation of *LTE*)



(b) *BCR* and *SCR* updation circuit

Fig. 5. Architectural support for ***db*** and ***cb*** in *B-2SC*

$LOAD(= s_0^{db})$ is high. *PGCC* is decreased by one when its input control line $DEC(= s_2^{db})$ is high. The output line **count not zero** (*CNZ*) is low when *PGCC* reaches zero indicating completion of ***db***. The $(x+1)$-bit register Q_{tot}^r stores the cumulative sum of charge drawn due to wake-up of first $(PGCC - \#component\_list + 1)$ *PG* components in $\#component\_list$. An $(x+1)$-bit adder *ADDQ* adds Q_α^r to Q_{tot}^r when the input control line $ADD(= s_1^{db})$ is high. An $(x+1)$-bit comparator *CMPQ1* compares Q_{tot}^r with *TSOCR* when the input control signal $CMP1(= s_2^{db})$ is high. There is an **insufficient charge flag** (*ICF*). It is set if the supercapacitors have insufficient charge to deal with the required wake-up. *ICF* is cleared when its input control line $CLR_{icf}(= s_0^{db})$ is high. *PGCC* is cleared when its input control line $CLR_{pgcc}(= ICF$ **and** $s_2^{db})$ is high. Bit BCR_σ of *BCR* is cleared if the input control line $CLR_{bcr\sigma}$ is high. Bit SCR_σ of *SCR* is cleared if the input control line $CLR_{scr\sigma}$ is high.

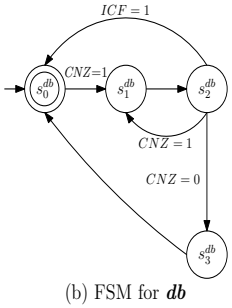
If $Q_{tot}^r > TSOCR$ then *CMPQ1* sets the *ICF*. *B* is connected to $C_0, C_1, \cdots, C_{n-1}$ and SC_0, SC_1 are charged. The input control line *CMP2* is high iff $ICF = 0, CNZ = 0$ and $s_2^{db} = 1$. This enables the comparator *CMPQ2* to find the maximum $SOCR_\sigma$ which is stored in *MSOC*. The supercapacitor with maximum charge flag (*SCMC*) holds the value of σ representing SC_σ. The comparator *CMPQ3* is enabled when its input control signal *CMP3* is high iff $ICF = 0, CNZ = 0$ and $s_3^{db} = 1$. If $Q_{tot}^r \leq MSOC$ then the output of *CMPQ3* **less than or equal to** line $LTE = 1$, otherwise $LTE = 0$. If $LTE = 1$ then the bits BCR_σ and $SCR_{1-\sigma}$ are set to while $BCR_{1-\sigma}$ and SCR_σ are reset to discharge SC_σ and charge $SC_{1-\sigma}$. If $LTE = 0$ then for simultaneous discharge of SC_0 and SC_1 BCR is assigned $(11)_2$ and *SCR* is assigned $(00)_2$.

C. Micro-operations for db and cb

<u>*db* $\#components, component\_list$</u>

s_0^{db} : $ICF \leftarrow (0)_2, Q_{tot}^r \leftarrow 0$
  $PGCC \leftarrow \#components$
s_1^{db} : $\alpha \leftarrow component\_list[\#components - PGCC]$
  $Q_{tot}^r \leftarrow Q_{tot}^r + Q_\alpha^r$
s_2^{db} : **if** $Q_{tot}^r > TSOCR$ **then**
    $PGCC \leftarrow 0, ICF \leftarrow (1)_2$
    $BCR \leftarrow (00)_2, SCR \leftarrow (00)_2$
    Goto s_0^{db} and **Stop**.
  end if
  $PGCC \leftarrow PGCC - 1$
  if $CNZ = (1)_2$ **then Goto** s_1^{db}.
  if $CNZ = (0)_2$ **then**
    $SOCR\sigma \leftarrow$ **max**$(SOCR0, SOCR1) | \sigma \in \{0,1\}$
    $MSOC \leftarrow SOCR\sigma, SCMC \leftarrow \sigma$
  end if
s_3^{db} : **if** $Q_{tot}^r \leq MSOC$ **then**
    $BCR_\sigma \leftarrow (1)_2, SCR_\sigma \leftarrow (0)_2$
    $BCR_{1-\sigma} \leftarrow (0)_2, SCR_{1-\sigma} \leftarrow (1)_2$
  else
    $BCR \leftarrow (11)_2, SCR \leftarrow (00)_2$
  end if
  Goto s_0^{db} and **Stop**.

(a) Micro-operations for ***db***

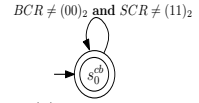


(b) FSM for ***db***

<u>*cb*</u>

s_0^{cb} : $BCR \leftarrow (00)_2, SCR \leftarrow (00)_2$

(c) Micro-operation for ***cb***



(d) FSM for ***cb***

Fig. 6. Micro-operations for ***db*** and ***cb*** in *B-2SC*

Figure 6(a) shows that in *EX* stage ***db*** for has four sequences – $s_0^{db}, s_1^{db}, s_2^{db}$ and s_3^{db}. Each sequence denotes a control signal generated by the finite state machine (FSM) shown in Fig. 6(b). The state transition in FSM takes place with a high clock input (CLK=1) conditionally/unconditionally denoted by a labeled/unlabeled edge. At a time one of the control lines is high to generate the corresponding control signal for performing the micro-operations belonging to a respective sequence.

At s_0^{db} *ICF* is reset, Q_{tot}^r is cleared and *PGCC* is loaded with *#components*. This followed by a transistion from s_0^{db} to s_1^{db} $(s_0^{db} \rightarrow s_1^{db})$. At s_1^{db}, α is obtained from *component_list* and Q_α^r is added to Q_{tot}^r followed by $s_1^{db} \rightarrow s_2^{db}$. At s_2^{db}, Q_{tot}^r is compared *TSOCR*. *ICF* is set and *PGCC* is cleared if Q_{tot}^r exceeds *TSOCR* leading to $s_2^{db} \rightarrow s_0^{db}$ indicating completion of ***db*** with *B* remaining connected. If Q_{tot}^r does not exceed *TSOCR* then *PGCC* is decreased by one. If *CNZ* is set then $s_2^{db} \rightarrow s_1^{db}$ occurs. At s_2^{db}, if $CNZ = 0$ then **max**(*SOCR0*, *SOCR1*) is obtained in *MSOC* and $\sigma \in \{0,1\}$ is obtained in *SCMC* followed by $s_2^{db} \rightarrow s_3^{db}$. At s_3^{db}, if $Q_{tot}^r \leq MSOC$ then BCR_σ and $SCR_{1-\sigma}$ are set to discharge SC_σ while $BCR_{1-\sigma}$ and SCR_σ are reset to charge $SC_{1-\sigma}$. If $Q_{tot}^r > MSOC$

then *BCR* is assigned $(11)_2$ and *SCR* is assigned $(00)_2$ for simultaneous discharge of SC_0 and SC_1.

The micro-operation in the *EX* stage of ***cb*** comprises of only one sequence s_0^{cb}. Figure 6(c) shows at s_0^{cb} *BCR* and *SCR* are assigned $(00)_2$. This is possible if $BCR \neq (00)_2$ **and** $SCR \neq (00)_2$ as shown in Fig. 6(d).

D. DB-CB scheduling in programs with PG instructions

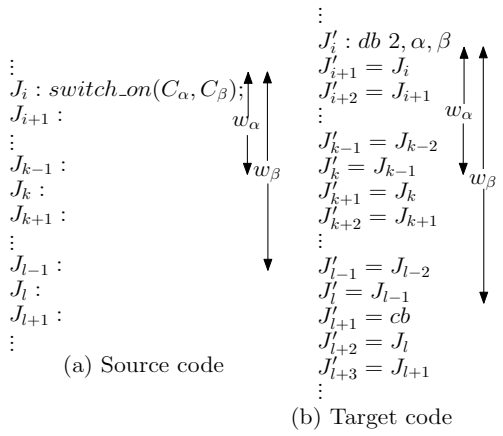


Fig. 7. Code Translation

The source code in Fig. 7(a) is translated to the target code with scheduled ***db-cb*** instructions in Fig. 7(b). This is done by means of **DB-CB Scheduling Algorithm** proposed in [2]. A ***db*** is inserted as immediate predecessor each *switch_on*. A ***cb*** instruction is inserted after completion of the wake-up of all the components involved in overlapped or non-overlapped wake-up. This ensures that the in-rush current is drawn from supercapacitor(s) and not from battery.

IV. Experiment and Results

To establish the efficacy of the proposed approach, simulations are carried out on **gem5** [14] architecture simulator. McPAT [16] is used for obtaining power values. **gem5** is configured with the instruction set and functional units (*FU*s) of the ARM Cortex-M4F processor [15]. The processor has seven *FU*s. Integer ALU is not power gated because it is used in majority of the instructions. The bits 0, 1, 2, 3, 4 and 5 of PGCR are the *PG* bits of Floating Point Divider, Floating Point Multiplier, Floating Point Adder, Integer Divider, Integer Multiplier and Barrel Shifter, respectively as shown in Fig. 8. The size of the instruction cache is considered to be 32 KB. The architectural support for ***db*** and ***cb*** are added to ARM Cortex-M4F.

McPAT is configured with the power model of ARM Cortex-M4F based on 32nm process technology, where the leakage power dissipation is almost 70% of the total power consumption. Here, the processor clock frequency $f_{clk} = 1.0$ GHz, and power supply voltage $V_{dd} = 0.9$ V. V_{th} of processor's n-MOS and p-MOS transistors are $V_{tn} = 0.18$ V and $V_{tp} = -0.18$ V, respectively. V_{th} of p-MOS transistors which act as header switches are -0.45 V. $I_{tol} = 200$ mA. Table II show the values of load capacitance ($c_l^{(\alpha)}$), maximum operating current ($I_{op}^{(\alpha)}$), w_α, peak I_α (I_α^{pk}) and Q_α^r for each C_α.

TABLE II
Values of $c_l^{(\alpha)}, I_{op}^{(\alpha)}, w_\alpha, I_\alpha^{pk}$ and Q_α^r

C_α	*fpdiv*	*fpmul*	*fpadd*	*idiv*	*imul*	*bshf*
$c_l^{(\alpha)}$(in nF)	6.58	5.9	3.89	2.24	1.81	0.8
$I_{op}^{(\alpha)}$(in mA)	17.24	15.47	12.84	9.63	8.12	4.72
w_α(in $cycles$)	32	30	24	18	16	10
I_α^{pk}(in mA)	185	177	146	112	102	72
Q_α^r (in Charge units)	942	864	672	563	497	355

A battery-supercapacitor simulator has been designed for both sole battery powered system comprising of *B* as well as for *B-SC* [2] and *B-2SC* systems. *B* is considered to be a Li-ion battery with $V_{batt} = 3.7$V and capacity of 2100 mAh. Considering 1 Coulomb as 10^{12} charge units, *B* stores a maximum of 7.56×10^{15} charge units. In *B-SC* [2], the supercapacitor has a supply voltage $V_{dd} = 0.9$V, a capacitance of $4.32nF$ and a storage capacity of 3892 charge units. In *B-2SC*, each SC_σ has a supply voltage $V_{dd} = 0.9$V, a capacitance of $2.16nF$ and a storage capacity of 1946 charge units where $\sigma \in \{0, 1\}$. The *CC* and DC-DC converter are also considered in these simulation environments. For Li-ion batteries the DC-DC converter scales down 3.7V to 0.9V. Its efficiency is considered as 90%. R_σ of the *CC* is considered to be $10k\Omega$. The load switch p-MOS transistor *T* is considered to have $V_{th} = -0.18$V and source voltage $V_S = V_{dd}$.

The GCC compiler for ARM Cortex-M4F [17] is extended to replace high level *switch_on* and *switch_off* instructions with equivalent ***pg pgcr_bit_vector*** instruction. The features leading to generation of basic *PG* (using [12], [13]) and translated *PG* code with ***db-cb*** are also added to the GCC compiler for ARM Cortex-M4F. The proposed techniques

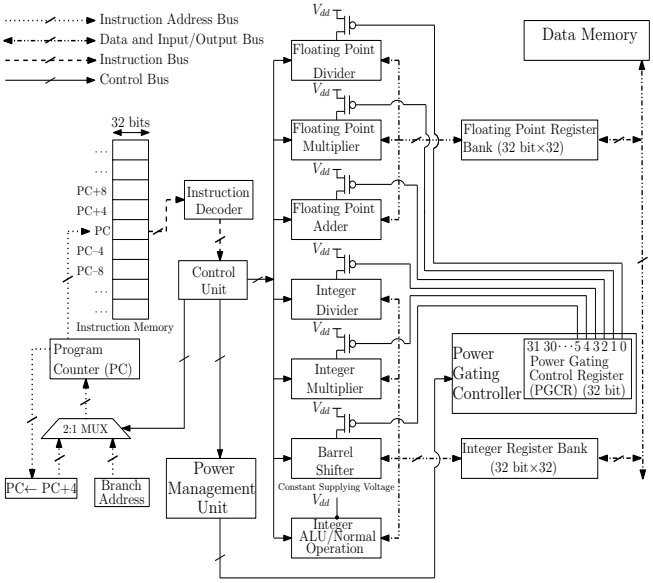


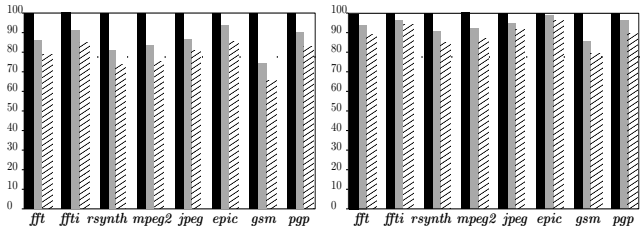
Fig. 8. A machine architecture model with *PG* control

are tested on MiBench [18] and MediaBench [19] benchmark programs as shown in Table III which contains the number of wake-up regions ($\#wk\_reg$) for each benchmark programs.

TABLE III
BENCHMARK DESCRIPTION

Program	*fft*	*ffti*	*rsynth*	*mpeg2*	*jpeg*	*epic*	*gsm*	*pgp*
Bench. Suite	MiBench	MiBench	MiBench	Media	Media	Media	Media	Media
Category	Telecomm	Telecomm	Office	Video	Image	Image	Speech	Crypto
$\#wk\_reg$	18	12	21	20	17	9	25	14

The benchmark programs are compiled using updated GCC compiler. The generated target code is executed on gem5 behaving as ARM Cortex-M4F processor. The performance values are generated by gem5. These performance values along with process technology and power related parameters of ARM Cortex-M4F act as input to McPAT which produces the power trace containing the values of peak, average, dynamic and leakage power. The values of the in-rush and load currents are obtained from the peak and average power values, respectively. These current values are converted to charge units. The charge units consumed by the basic *PG* programs executed on system powered by *B* and translated battery aware *PG* programs executed on systems powered by *B-SC* and *B-2SC* are supplied to the battery-supercapacitor simulator for computing C-rate.



(a) Normalized average C-rate (in %) (b) Normalized performance (in %)

Fig. 9. Comparison of average C-rate and performance wrt delay

The experimental results are shown in Fig. 9. The programs with *PG* instructions running on the system powered by *B* and their translated versions running on systems powered by *B-SC* [2] and *B-2SC* are compared in terms of C-rate and performance with respect to delay. The normalized average C-rate of the original program and translated programs are compared in Fig. 9(a). The normalized average C-rate values with respect to the original programs are obtained using the equation (1)

$$NORM\_CRATE_v = 100 - \frac{(CRATE_v - CRATE_B)}{CRATE_B} \times 100 \quad (1)$$

where v denotes the version of original/translated program and $v \in \{$*B,B-SC,B-2SC*$\}$. $NORM\_CRATE_v$ and $CRATE_v$ are the normalized average C-rate (in %) and average C-rate for version v of a program, respectively. Translated programs having more $\#wk\_reg$ achieve higher reduction in C-rate because C-rate $\propto \#wk\_reg$. The normalized performance with respect to delay of the original and translated programs are compared in Fig. 9(b). The normalized performance values with respect to the original programs are obtained from the equation (2)

$$NORM\_PERF_v = 100 - \frac{(T_v - T_B)}{T_B} \times 100 \quad (2)$$

where $NORM\_PERF_v$ and T_v are the normalized performance (in %) and delay (in cycles) for version v of a program, respectively. Delay of a translated program $\propto$ number of ***db-cb*** instructions $\propto \#wk\_reg$. For a successful disconnection of *B* the ***db*** in *B-2SC* consumes one clock cycle more than the ***db*** in *B-SC* [2]. Hence the translated programs running on *B-2SC* powered system experience higher delay. When compared with the system with *B*, *B-SC* achieves 7-25% reduction in C-rate at the cost of 1-12% degradation in performance, while *B-2SC* achieves 14-34% reduction in C-rate at the cost of 3-20% degradation in performance.

V. CONCLUSION

The present work introduces scheduling in a *B-2SC* system for reducing battery C-rate to increase battery lifetime in systems with *PG*. Two battery control instructions ***db*** and ***cb*** are introduced with architectural support. The insertion of ***db*** and ***cb*** in a program with *PG* instructions is explained. *B-2SC* facilitates simultaneous charging/discharging of SC_σ and $SC_{1-\sigma}$ where $\sigma \in \{0, 1\}$. Hence *B-2SC* achieves higher reduction in C-rate than *B-SC* [2] at the cost of higher performance degradation. The future work will investigate to reduce the delay of ***db*** followed by a detailed study on the scalability of multiple supercapacitors and its effect on battery lifetime as well as performance.

REFERENCES

[1] R. A. Dougal, S. Liu, and R. E. White, "Power and Life Extension of Battery-Ultracapacitor Hybrids," *IEEE Trans. on Components and Packaging Technologies*, Vol. 25, No. 1, pp. 120-132, March 2002.

[2] S. Pyne, "Scheduling of Hybrid Battery-Supercapacitor Control Instructions for Longevity in Systems with Power Gating," *in Proc. ACM/IEEE ISLPED*, pp. 45:1-45:6, 2018.

[3] Q. Wu, Q. Qiu and M. Pedram, "An interleaved dual-battery power supply for battery operated electronics", *in Proc. ASP-DAC*, pp.387-390, 2000.

[4] L. Benini, G. Castelli, A. Macii, E. Macii and R. Scarsi, "Battery-Driven Dynamic Power Management of Portable Systems", *in Proc. ACM/IEEE ISSS*, pp. 25-33, 2000.

[5] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncin and R. Scarsi, "Extending Lifetime of Portable Systems by Battery Scheduling," *in Proc. DATE*, pp. 197-201, 2001.

[6] C. Chiasserini and R. Rao, "Energy efficient battery management," *IEEE J. Sel. Areas in Commun.*, Vol. 19, No. 7, pp. 1235 - 1245, 2001.

[7] V. Rao, G. Singhal, A. Kumar, and N. Navet, "Battery model for embedded systems," *in Proc. Intl. Conf. VLSI Design and Embedded Systems Design*, 2005, pp. 105-110.

[8] D. Shin, Y. Kim, J. Seo, N. Chang, Y. Wang, and M. Pedram. 2011. "Battery-supercapacitor hybrid system for high-rate pulsed load applications," *in Proc. DATE*, pp. 1-4, 2011.

[9] P. Rong and M. Pedram, "Battery-Aware Power Management Based on Markovian Decision Processes," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 25, No. 7, pp. 1337-1349, 2006.

[10] http://www.richtek.com/en/Design Support/Technical Document/AN024

[11] S. Narayanaswamy, S. Schlueter, S. Steinhorst, M. Lukasiewycz, S. Chakraborty and H. E. Hoster, "On Battery Recovery Effect in Wireless Sensor Nodes," *ACM Trans. Design Autom. Electr. Syst.*, Vol. 21, No. 4, 2016.

[12] Y. P. You, C. Lee and J. K. Lee, "Compilers for Leakage Power Reduction," *ACM Trans. Design automation of Electronic Syst.*, Vol. 11, No. 1, pp. 147-164, 2006.

[13] S. Roy, S. Katkoori and N. Ranganathan, "A Framework for Power-Gating Functional Units in Embedded Microprocessors," *IEEE Trans. VLSI Syst.*, Vol. 17, No. 11, pp. 1640-1649, 2009.

[14] http://gem5.org/Main_Page

[15] http://infocenter.arm.com

[16] http://www.hpl.hp.com/research/mcpat/

[17] https://developer.arm.com/open-source/gnu-toolchain/gnu-rm

[18] http://vhosts.eecs.umich.edu/mibench//

[19] http://mathstat.slu.edu/~fritts/mediabench/