Simultaneous Scheduling and Binding for Power Minimization During Microarchitecture Synthesis

Aurobindo Dasgupta and Ramesh Karri
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003
dasgupta@ecs.umass.edu *karri@ecs.umass.edu*

ABSTRACT – Sub-micron technologies and the increasing size and complexity of integrated components have aggravated the effect of long interconnects and buses, compared to that of gates, on the overall performance, and energy of systems [13]. Consequently, we propose a RT level design technique to reduce the energy dissipated in switching of the buses ($\approx$ 40% of the on-chip power [13]) in the synthesized microarchitectures. This is accomplished by judiciously binding/scheduling data transfers in a Control Data Flow Graph (CDFG) onto buses in the design. The algorithm considers (i) correlations between data transfers, (ii) constraints on system performance, and (iii) constraints on the number of buses. Simulations on benchmarks show that best-energy designs are up to 75% energy-efficient vis-a-vis the worst-energy designs. Further, best-energy designs are up to 45% more energy-efficient than the best-delay designs.

1. INTRODUCTION

Proliferation of battery-operated applications that demand intensive computation in a portable environment necessitates the design of power efficient circuits. The dominant source of power dissipation in digital CMOS circuits is the dynamic power,P [4], given below:

$$P = f \cdot V_{dd}^2 \cdot \left(\sum_{i=1}^{N} p_i \cdot C_i \right) \quad (1)$$

In equation 1, V_{dd} is the supply voltage, f is the clock frequency, N is the number of nodes in the circuit, p_i is the probability that the ith node toggles in a clock cycle and C_i is the capacitance of the ith node.

Power optimization techniques involving the reduction of V_{dd} to reduce power are presented in [4]. However, decreasing V_{dd} reduces current drive capability, thus decreasing the circuit throughput. Therefore, such techniques are typically supplemented with throughput enhancement mechanisms such as duplication of hardware and pipelining [1]. Reduction of power by minimizing the estimated switching activity [7, 9], p_i, has been investigated at the logic level by [10, 14] and at the layout level by [5]. Power minimization in the algorithmic level has been studied by [1] using computational transformations such as operation reduction, operation substitution, word length reduction and transformation ordering. Similar techniques were used for low power at the RT level using flow graph restructuring, retiming and pipelining [8]. In [2], an allocation method during behavioral synthesis for low power is described. This allocation technique reduces the power dissipated in registers and is based on a point-to-point interconnection model. In [12], coding of signals is used to optimize power dissipation in buses/IO lines. This result in increased bus width and number of I/O pins.

Sub-micron feature sizes have resulted in a significant percentage ($\approx$ 40% of the on-chip power [3, 13]) of power to be dissipated on the buses, leading to an increased focus for greater savings for power at the algorithmic level and RT level of design. Constraints on available silicon area, forces signals from different functional units to be multiplexed on to a single bus. Multiplexing of different signals onto these highly capacitive buses tends to increase switching activity thereby increasing power dissipation [11]. This motivated us to investigate methods to reduce power dissipation on the highly capacitive buses.

The average energy, E, dissipated by a circuit is given by $E = P \times T$, where P is the average power and T is the delay of the circuit. The energy dissipated determines battery life and therefore, energy minimization is targeted in power-related research. Furthermore, the intrinsic reliability of a circuit due to failures caused by electromigration, hot electron effect, ground bounce and excessive voltage drop across power buses is aggravated by sub-micron technology[6]. These failures

are directly related to the charge drawn by the circuit. Since the energy dissipated in a circuit is proportional to the charge drawn by it, minimizing energy dissipation indirectly improves the intrinsic reliability of a circuit as well. We therefore focus on **energy** dissipation in a circuit.

In this paper, we show how to synthesize energy-optimized circuits at the RT level by multiplexing the signals onto buses such that the switching activity[11], on these buses is minimized. Reducing the switching activity on buses has been shown to be equivalent to reducing the energy[9]. Apart from the fast estimation of data transfer switching activity (taking in to account signal correlation), the width of the buses and constraints on latency and area, are also considered.

2 MOTIVATION & TERMINOLOGY

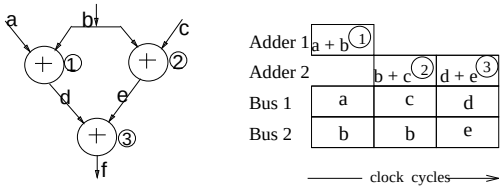

Figure 1: (a) An example CDFG. (b) A possible schedule for the CDFG.

Consider an example control Data Flow Graph (CDFG) with three add operations (1, 2 and 3) and 6 data transfers (a,b, c, d, e and f) in figure 1(a). Every add operation is assumed to require one clock cycle. One possible schedule of the CDFG is shown as a gantt chart in figure 1(b). The total execution time of this schedule is three clock cycles. The microarchitecture corresponding to the gantt chart of figure 1(b), is shown in figure 2. Two adders and two buses are used to implement this schedule. Data transfers a, c and d are implemented in successive clock cycles on bus 1. Similarly, data transfers b and e are implemented successively on bus 2. The sequence of data transfers on the buses significantly influences the energy dissipated on the buses.

The **average switching activity** on a bus at time instant t, is the average number of lines on the bus that have either a $0 \rightarrow 1$ or a $1 \rightarrow 0$ transition at t. The switching activity on a bus is also an indicator of the average number of lines of the bus that toggle or the average number of bit flips on the bus. The total energy dissipated on a bus is proportional to the switching activity on the bus [9].

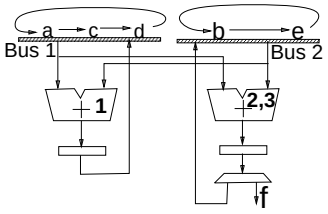

Figure 2: The microarchitecture corresponding to the schedule in figure 1(b)

The total energy dissipated in the buses can be computed as follows. Let data transfers i and j be consecutively scheduled on bus k, and let SA^k_{ij} be the average switching activity when bus k changes from the data transfer i to the data transfer j. If p_l is the transition probability of the l^{th} line in the bus at the time of transition from i to j on bus k then,

$$SA^k_{ij} = \sum_{\forall line\ l \in k} p_l \qquad (2)$$

	a	b	c	d	e	f
a	3.99	3.97	4.02	3.98	3.95	3.96
b	3.97	3.97	4.00	3.98	4.07	4.03
c	4.02	4.00	3.94	3.94	3.96	4.03
d	3.98	3.98	3.94	3.99	4.07	3.93
e	3.95	4.07	3.96	4.07	4.02	4.06
f	3.96	4.03	4.03	3.93	4.06	3.97

Table 1: Values of SA_{ij} for all i, j

Values of SA^k_{ij} for all possible data transfers i and j, for the CDFG in figure 1 are listed in table 1.These values are computed assuming that (i) the primary inputs are independent of each other, both spatially (primary inputs used in the same computation) and temporally(primary inputs used for two successive data computations), (ii) the primary inputs have a logical value of 1 with a probability 0.5 and (iii) the bus width is 8. The method of computing these values is explained in detail in section 3.1.

From table 1, SA^k_{ac} is 4.02. This means that an average of 4.02 lines out of the 8, toggle when the signal value of bus k changes from a to b. In a nutshell, the value of S^k_{ij} depends on (i) the probabilistic signal values of the primary inputs, (ii) the bus widths and (iii) the input CDFG.

The values of SA_{ij}^k, for all data transfers i and j successively scheduled on bus k, determines the total average switching activity, SA_{ij}^k, and is given by equation 3.

$$SA^k = \sum_{\forall (i \rightarrow j)\ transitions\ on\ k} SA_{ij}^k \quad (3)$$

From equation 3, the switching activity on bus 1, in figure 2 is $SA^1 = SA_{ac}^1 + SA_{cd}^1 + SA_{da}^1 = 11.94$. Similarly, the switching activity on bus 2 is $SA^2 = SA_{be}^2 + SA_{eb}^2 = 8.14$.

The total energy dissipated on the buses is proportional to the expected switching activity on all buses (denoted by SA) and can be computed using equation 4.

$$SA = \sum_{\forall k\ that\ is\ used} SA^k \quad (4)$$

For our example, $SA = SA^1 + SA^2 = 20.08$. SA also determines the average number of transitions ($0 \rightarrow 1$ and $1 \rightarrow 0$) on all buses during one CDFG computation.

The proposed method for energy-efficient synthesis does not attempt to reduce the energy dissipated in the control circuit (multiplexors), although the energy dissipated may reduce in the final design. The attempt is not made because the number of lines required for the control circuitry is much smaller than the number of lines in the buses, and is usually much shorter (smaller capacitance) than the buses.

3. THE ALGORITHM

Motivated by the observations in section 2, we propose an algorithm for simultaneous scheduling and binding to determine a sequence for the data transfers onto the buses such that the switching activity, SA is minimized. The algorithm consists of two stages namely (a) activity determination and (b) simultaneous scheduling and binding, the details of which are explained below.

3.1 ACTIVITY DETERMINATION

The values of SA_{ij}^k are computed by repeated simulation of the CDFG. These activities are required in the subsequent stage of scheduling and binding. The activity determination stage accepts as its inputs (i)the CDFG, (ii)the bus width and (iii)the signal probability of the primary inputs.

As mentioned earlier in section 2, signal probabilities of the primary inputs are assumed to be spatially and temporally independent, and are used to generate the values for the primary inputs. For each set of primary inputs, the signal values for all data transfers in the CDFG are computed by simulating the CDFG. For every possible pair of data transfers i and j in one CDFG simulation, the number of bit flips during the transition $i \rightarrow j$ is calculated by $\oplus$-ing the corresponding bits in the signal values of i and j. Simulation of the CDFG is then repeated for a new set of primary input values. The CDFG simulations are terminated when the average number of bit flips for all data transfer pairs converge.

For DSP applications, where the functional units are mostly adders and multipliers, such simulations can be done very quickly. This is because, only simple operations such as additions and multiplications, are required to generate the values for the data transfers. Also, the number of bit flips during a transition between a pair of data transfers is determined using simple $\oplus$ operations. The average number of bit flips during a transition between pairs of data transfers, SA_{ij}^k, is updated after every simulation, is found to converge very quickly. Suppose n is the number of operations(or data transfers) in the CDFG then n^2 is the number of possible data transfer pairs. If S is the number of simulations required for convergence, then the total time required to compute SA_{ij}^k for all possible data transfer pairs i and j is given by $O(S \cdot n^2)$.

Since it is inexpensive to perform these simulations, it is feasible to simulate the CDFG to compute SA_{ij}^k by taking into account the correlations between the signals i and j. Therefore, for a given schedule on a microarchitecture, the values of SA_{ij}^k, can also be used for a quick estimation of the energy dissipated on buses.

3.2 SIMULTANEOUS SCHEDULING & BINDING

This stage accepts as its inputs (i) the CDFG, (ii) the values of SA_{ij}^k computed in the activity determination stage, (iii) the number of buses and functional units available for binding and, (iv) the execution times of the operations on the available functional units. The algorithm then synthesizes an energy-efficient microarchitecture subject to the constraints on the area (number of functional units and buses) and delay. The following shows the overall structure of the algorithm :

---

1. Select an initial value SA_{curr}.
2. **While** stopping condition is FALSE
3. **Repeat** for a certain number of iterations

   Repeat until ($T_{max} < D$) or (for a certain number of iterations).

   /* Module Selection */

Repeat until $\sum_{\forall i \in M} Area_i < AREA$

4. Select a module set, M.

End Repeat.

/* Binding and Sequencing */

Repeat until $(T_{max} < D)$ or (for a certain number of iterations).

5. Randomly bind and sequence data transfers onto buses.

Compute SA_{new}, (using equation 4).

/* Scheduling */

Repeat until $(T_{max} < D)$ or (for a certain number of iterations).

6. Schedule/reschedule operations onto functional units and data transfers onto buses.

End Repeat.

End Repeat.

End Repeat.

7. if $SA_{new} < SA_{curr}$ then $SA_{curr} \leftarrow SA_{new}$
else $SA_{curr} \leftarrow SA_{new}$ with a given probability.

End Repeat

End While

Let $AREA$ be the user-specified area upper bound for the microarchitecture to be synthesized. The area of the microarchitecture is determined by the number of functional units and buses used in the design. Similarly, let D be the user-specified delay upper bound. The delay is determined by the total number of clock cycles required to execute the CDFG on the synthesized microarchitecture, and is denoted by T_{max}. For simplicity, and without loss of generalization, the execution time for every operation on a functional unit is assumed to be no more than one clock cycle.

In this algorithm, the objective to be minimized is the total switching activity on the buses denoted by SA. This is equivalent to minimizing the total energy dissipated on the buses. Initially, a microarchitecture is synthesized ensuring that the area and performance constraints are met. Subsequently, this microarchitecture is incrementally refined to optimize SA, the energy dissipated on the buses, without violating the area or performance constraints. Starting with an arbitrarily high value for the switching activity, SA_{curr} (step 1), simulated annealing is used to arrive at an energy efficient microarchitecture.

In order to satisfy the area constraint, the total area of the modules used should be less than $AREA$. In step 4, a module set, comprising of functional units and buses. The data transfers in the CDFG are then mapped and temporally ordered on the buses (Step 5). A temporal order determines only the sequence of implementation of the data transfers onto the buses and not the schedule(actual clock cycles when the data transfers takes place) of the data transfers. From the temporal order of data transfers on the buses and equation 4, the switching activity on the buses, SA, is computed. For example, in figure 2, the sequence of the data transfers $a \rightarrow c \rightarrow d$ and $b \rightarrow e$ is sufficient to determine the value of SA to be 20.08.

The CDFG is then scheduled (Step 6) to determine the actual clock cycles during which the data transfers are implemented on the buses and operations executed on the functional units. It is during this step that the total delay, T_{max}, of the synthesized microarchitecture is determined. If a schedule does not meet the delay constraint a new schedule is generated by implementing(executing) the data transfers(operations) such that the data transfer sequence on the buses is not altered. Other designs with different SAs are investigated by randomly remapping and resequencing data transfers onto buses and then scheduling them until no further improvement in SA is obtained.

All designs searched were synthesized using a fixed module set. Additional microarchitectures are investigated by altering(step 4) the module set. The module set is also altered, when despite several remappings and resequencings, the performance constraint remains unsatisfied.

The design procedure starting from module selection(step 4) is repeated in a simulated annealing loop. A *move* in this annealing routine is defined by the creation of a new module set. It is accomplished by adding or removing modules from the module set. The number of modules added or removed in a move is an indicator of the possible qualitative difference of the designs constructed from the module sets. This number decreases with the number of designs investigated during the annealing routine. Ultimately, an energy-efficient design for the synthesized microarchitecture which satisfies the cost and performance constraints, is obtained.

4. RESULTS

The results of the algorithm on the benchmark examples are summarized in Table 2. The effect of varying the number of buses on the energy dissipated in a circuit is shown for each of the three benchmarks. In Table 2, the third column contains the energy (measured as switching activity) and delay (measured as number of clock cycles) values for performance optimized microarchitectures. The fourth column, on the

Filter Examples	# of buses	Perf. SA/D	Energy SA/D	% Change SA/Del
AR	1	68/11	28/13	58.8/-15.3
	2	82/8	36/14	43.9/-42.8
	3	80/8	32/12	60.0/-33.0
	4	78/9	44/11	43.5/-18.1
	5	80/8	33/14	58.7/-42.8
	6	70/9	40/14	42.8/-35.7
FIR	1	50/9	19/13	62.0/-30.7
	2	50/11	19/12	62.0/-8.3
	3	52/8	29/13	44.2/-38.4
	4	69/8	37/11	46.3/-27.2
	5	60/7	33/11	45.0/-36.3
	6	44/8	25/12	42.2/-33.3
Elliptic	1	81/16	58/24	28.3/-33.3
	2	109/15	79/22	27.5/-31.8
	3	114/15	80/21	29.8/-28.5
	4	126/15	85/23	32.5/-34.7
	5	125/15	67/23	46.4/-34.7
	6	120/15	73/24	39.1/-37.5
Average				45.1/-31.24
Std. Dev.				11.1/8.9

Table 2: Results using the benchmarks

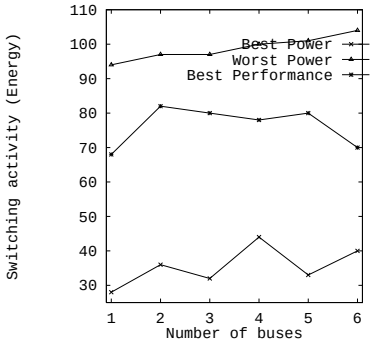

Figure 3: Energy dissipation vs. number of buses used for the AR Filter.

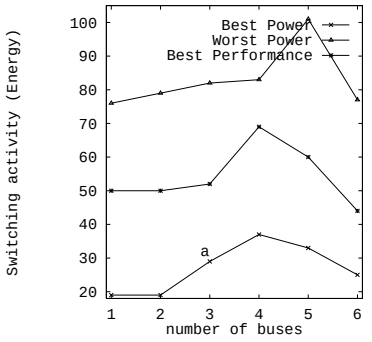

Figure 4: Energy dissipation vs. number of buses used for the FIR Filter.

other hand, contains the energy and delay values for energy optimized microarchitectures.

Percentage improvement of the energy dissipated in the synthesized energy-efficient microarchitecture and the synthesized performance-optimized microarchitecture is shown in column 5. The average improvement in the energy dissipated is found to be 45.1%. This indicates the extent of improvement in energy possible if there were no performance bounds. The percentage performance which was compromised to achieve an energy-efficient microarchitecture design is also shown in column 5. The average decrease in performance is found to be 31.8%. For a feasible performance bound, our algorithm is capable of synthesizing an energy-efficient microarchitecture satisfying this performance bound.

For the AR and FIR filter benchmarks, a plot of the energy dissipated against the number of buses used in the synthesized microarchitectures is shown in figures 3 and 4. Every point on the plot corresponds to a synthesized microarchitecture. For each figure, the three lines correspond to the best energy, worst energy and best performance microarchitectures, respectively. The considerable savings in energy that is possible using our method is evident from these plots. From these graphs, a significant difference in the dissipated energy for an energy-optimized microarchitecture and for a microarchitecture with the worst energy (selected from the solution space searched by the algorithm) can be seen. In fact, the average percentage difference is found to be 61.2%, which is an indicator of the large energy range of designs possible by altering the sequence of data transfers on buses. Thus, the scope for energy reduction using this method is considerable.

An example of an energy-efficient microarchitecture for an FIR filter is represented by point "a" in figure 4. This microarchitecture is synthesized using 3 buses. The synthesized microarchitecture is shown in figure 5.

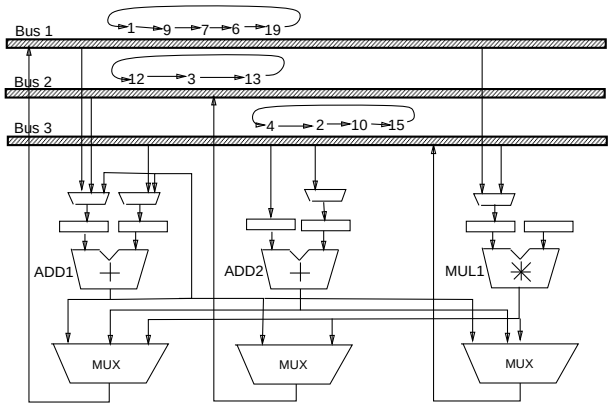

Figure 5: The data path for the best energy solution corresponding to point a in figure 4.

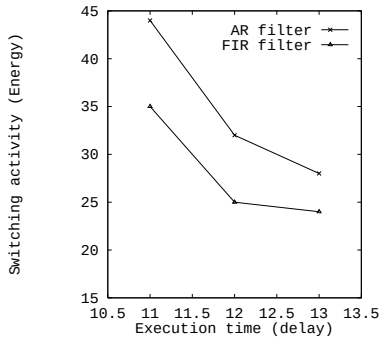

Figure 6: Energy dissipation vs. delay for AR and FIR filters

In figure 6, the plots of the average energy dissipated for a CDFG computation against the execution time (delay) for the synthesized microarchitectures, is shown. The points represent microarchitectures synthesized for energy-efficiency subject to performance constraints. From the plots it can be concluded that the energy dissipated on the buses for the synthesized microarchitectures decreases when the execution time of the microarchitecture increases. This is because a rearrangement of data transfers on the buses, done in an attempt to save energy may result in an increase in the total execution time.

5. CONCLUSION

We have presented a method to synthesize low-power microarchitectures. As a consequence of scaling, most of the power in a chip is dissipated on the buses. We have demonstrated that significant savings in power could be obtained by suitably binding and scheduling the data transfers of a CDFG onto buses so that the switching activity on these buses is minimized. Comparisons of the energy dissipated on benchmarks between the best-energy and worst-energy microarchitectures synthesized, demonstrated that an average of 61% improvement can be achieved. Comparisons between the energy dissipated in an energy-optimized and delay-optimized design indicated an average improvement of 45.1%.

References

[1] A. Chandrakasan, M. Potkonjak, J. Rabaey and R. Brodersen, "HYPER-LP: A system for power minimization using architectural transformations", ICCAD, pp 300-303, Nov 1992.

[2] A. Raghunathan, N. K. Jha, "Behavioral Synthesis for Low Power", ICCD, Nov 1994.

[3] H. B. Bakoglu, "Circuits, Interconnection and packaging for VLSI", Addisson Wesley, 1990

[4] A. Chandrakasan, T. Sheng, and R. W. Brodersen, "Low Power CMOS Digital Design",Journal of Solid State Circuits, pp 473-484, April 1992.

[5] K. Chao and D. F. Wong, "Low Power Considerations in Floorplan Design", IWLPD , pp. 45-50, 1993.

[6] A. Deng, "Power Analysis for CMOS/BiCMOS Circuits", pp 3-8, IWLPD, 1994.

[7] A. Ghosh, S. Devadas, K. Kuetzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits", 29th DAC, pp 253-259, 1992.

[8] L. Goodby, A. Orailoglu, P. Chau, " High-Level Synthesis for Low Power Design", ICCD 1994.

[9] F. N. Najm, "Transition Density, A Stochastic Measure of Activity in Digital Circuits", 28th DAC, pp 644- 649, 1991.

[10] K. Roy and S. Prasad, " SYCLOP : Synthesis of CMOS logic for low power application", Proc ICCD, October 1993.

[11] M. A. Schuette and J. R. Barr, "Embedded Systems Design for low energy consumption", ICCAD , pp 534-540, Nov 1994.

[12] M. R. Stan and W. Burleson, "Limited weight codes for low-power I/O", IWLPD , pp 209-214, April 1994.

[13] C. Svenson and D. Liu, "A power estimation tool and prospects of power savings in CMOS VLSI chips",Proc IWLPD, pp 171-176, April 1994.

[14] C. Tsui, M. Pedram and A. M. Despain, "Technology Decomposition and Mapping Targeting Low Power Dissipation", DAC, pp. 68-73, 1993.