LOW-POWER RECONFIGURABLE SIGNAL PROCESSING VIA DYNAMIC ALGORITHM TRANSFORMATIONS (DAT)

Manish Goel and Naresh R. Shanbhag

Coordinated Science Lab./ECE Department,
Univ. of Illinois at Urbana-Champaign,
1308 W. Main Street, Urbana IL-61801.
E-mail : [mgoel,shanbhag]@uivlsi.csl.uiuc.edu

ABSTRACT

Presented in this paper are dynamic algorithm transformations (DAT) for systematic design of *reconfigurable computing* engines. These techniques allow dynamic alteration of algorithm properties in response to input non-stationarities. The input is modeled as a set of states with an underlying probability distribution, $\mathcal{P}_{\mathbf{S}}$. For each input state $\mathbf{s}$, a signal monitoring algorithm **SMA** computes a power-optimal configuration for the signal processing algorithm **SPA** block. A fraction α of the **SPA** block is hardwired and the remaining $(1-\alpha)$ is reconfigurable. Similarly, the **SMA** block computation is partitioned into a fraction β for the memory and the remaining $(1-\beta)$ for the datapath. For the given input state distribution, the optimal values of α (α_{opt}) and β (β_{opt}) are determined. It is shown that for frequency selective filtering, the power savings of 35%-45% can be achieved by DAT-based reconfigurable system as compared to the traditional design based on the worst-case scenario.

1. INTRODUCTION

Algorithm transformation techniques [7] such as strength-reduction [9], reduced complexity VQ [10], block processing and associativity have been employed to design low-power and high-throughput systems. These transformations are referred to as *static algorithm transformations* (SAT), because these are applied during the algorithm design phase assuming a worst-case scenario and their implementation is time-invariant. Most real-life signal environments are non-stationary and hence significant power savings can be expected if the algorithm and architecture can be dynamically tailored to the input. This is the basis for the recent interest in *adaptive computing systems (ACS)* [1], where run-time reconfigurability is a key feature. A related approach is that of *data-driven signal processing* [3], where the algorithm workload and the voltage supply are varied in real-time to optimize the power dissipation.

In [4], we proposed *dynamic algorithm transformations* (DAT) as another approach to data-driven reconfigurable signal processing [8]. The block level diagram of the DAT-based system is shown in Fig. 1. The **SPA** block (see Fig.1) implements the main signal processing function, which could vary over time. On the other hand, the **SMA** block decides the instant and the extent of modification to the **SPA** block so as to optimize the average energy dissipation while maintaining a pre-specified level of output performance.

In this paper, we define input states and an underlying probability distribution for these states. For each of the states, the **SMA** block computes an optimum-energy configuration for the **SPA** block. The **SPA** block (see Fig. 1) is partitioned into a hardwired **HSPA** and a reconfigurable **RSPA** block. The **SMA** block (Fig. 1) is partitioned into datapath **HSMA** and memory **MSMA** blocks. Energy-optimum partitions are then obtained by minimizing the total DAT energy dissipation. This approach is applied to frequency selective filters to get significant energy savings.

This work was supported by DARPA contract DABT63-97-C-0025 and NSF CAREER award MIP-9623737.

2. DYNAMIC ALGORITHM TRANSFORMATIONS (DAT)

In this section, we present DAT as a systematic method for designing a reconfigurable DSP system.

2.1. Probabilistic Setup

We present a probabilistic setup for design of DAT-based reconfigurable systems.

Definition 1 : *The* <u>*input state*</u> $\mathbf{s}$ *at any instant* nL, $\mathbf{s}(nL) \in \mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{N_s}\}$ *is an algorithm dependent parameter defined for a block of* L *input samples* $x(nL), x(nL-1), \cdots, x(nL-L+1)$.

The set $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{N_s}\}$ is usually predefined and is algorithm dependent. For example, one definition of the input state $\mathbf{s}(nL)$ could be $E_{sb,x}(nL)$, where $E_{sb,x}(nL)$ is the stopband energy computed via L input samples $x(nL), x(nL-1), \cdots, x(nL-L+1)$. The state $\mathbf{s}$ can be detected after L sample periods from a block of L input samples.

We assume that the input state $\mathbf{s}$ takes values from $\mathcal{S}$ according to a probability mass function $\mathcal{P}_{\mathcal{S}}(\mathbf{s})$ defined as

$$\mathcal{P}_S(\mathbf{s}_i) = Probability(\mathbf{s} = \mathbf{s}_i), \quad 1 \le i \le N_s \qquad (2.1)$$

where $\mathcal{P}_S(\mathbf{s}_i) \ge 0$, $\sum_i^{N_s} \mathcal{P}_S(\mathbf{s}_i) = 1$ and N_s is the number of states.

Definition 2 : *The* <u>**SPA** *configuration*</u> $\mathbf{C} = [c_1, \cdots, c_N]$ *is a vector of signals which reconfigure the* **SPA** *block thus changing the algorithm being implemented.*

For example, $\mathbf{C} = [h_1, h_2, \cdots, h_N]$ is a possible **SPA** configuration for an N-tap FIR filter, where h_i are B-bit coefficients of the FIR filter.

Definition 3 : *The* <u>*optimum* **SPA** *configuration*</u> $\mathbf{C}_{opt}(\mathbf{s})$ *is an* **SPA** *configuration such that the* **SPA** *block dissipates minimum energy while achieving specified algorithm performance for input state* $\mathbf{s}$.

After every L samples, the **SMA** block will be employed to compute $\mathbf{C}_{opt}(\mathbf{s}(nL))$, where $\mathbf{s}(nL)$ can be any of the N_s states defined by $\mathcal{S}$. In the traditional static framework, $N_s = 1$, $\mathbf{s}$ is the worst-case input state and $\mathbf{C}_{opt}(\mathbf{s})$ is the worst-case **SPA** configuration.

Definition 4 : *The* <u>*transition probability*</u> $\mathcal{P}_T(\mathbf{s}_i)$ *for state* $\mathbf{s}_i$ *is defined as*

$$\mathcal{P}_T(\mathbf{s}_i) = \mathcal{P}_S\left(\mathbf{s}(nL) = \mathbf{s}_i, \mathbf{s}(nL-L) \ne \mathbf{s}_i\right). \qquad (2.2)$$

It can be seen that $\mathcal{P}_T(\mathbf{s}_i)$ is also the probability that reconfiguration needs to take place. It can be shown that for temporally uncorrelated $\mathbf{s}(nL)$, $\mathcal{P}_T(\mathbf{s}_i)$ increases with $\mathcal{P}_S(\mathbf{s}_i)$.

2.2. Energy Dissipation

Let α be the fraction of **SPA** block which is hardwired, and let β be the fraction of the **SMA** block in the form of memory. Then, the average energy dissipation per sample for a DAT-based system, $\mathcal{E}_{DAT}(\alpha, \beta)$ is given by,

$$\mathcal{E}_{DAT}(\alpha,\beta) = \mathcal{E}_{SPA}(\alpha) + \mathcal{E}_{SMA}(\alpha,\beta), \qquad (2.3)$$

where $\mathcal{E}_{SPA}(\alpha)$ and $\mathcal{E}_{SMA}(\alpha,\beta)$ is the average energy dissipated per sample by the **SPA** and the **SMA** respectively. The $\mathcal{E}_{SPA}(\alpha)$ can be further decomposed as:

$$\mathcal{E}_{SPA}(\alpha) = \mathcal{E}_{HSPA}(\alpha) + \sum_{i=1}^{N_s} \mathcal{E}_{RSPA}\left(\mathbf{C}_{opt}(\mathbf{s}_i)\right)\mathcal{P}_S(\mathbf{s}_i), \qquad (2.4)$$

where $\mathcal{E}_{HSPA}(\alpha)$ is the average energy dissipated per sample by the hardwired **HSPA** block. $\mathcal{E}_{RSPA}(\mathbf{C})$ is the average energy dissipated per sample by the reconfigurable **RSPA** block in configuration $\mathbf{C}$.

Similarly, the energy dissipation of the **SMA** block can be computed. Note that $C_{opt}(\mathbf{s}_i)$ for $\mathbf{s}_i \in \mathcal{S}_0 = \{\mathbf{s}_1, \cdots, \mathbf{s}_{\beta N_s}\}$ have been precomputed and stored in memory **MSMA**, while $C_{opt}(\mathbf{s}_i)$ for $\mathbf{s}_i \notin \mathcal{S}_0$ are computed online by employing **HSMA** block. The average energy dissipation per sample for the **SMA** block, $\mathcal{E}_{SMA}(\alpha,\beta)$ is given by

$$\mathcal{E}_{SMA}(\alpha,\beta) = \mathcal{E}_{DSMA} + \frac{1}{L}\left[\sum_{\mathbf{s}_i \in \mathcal{S}_0} \mathcal{P}_T(\mathbf{s}_i)\mathcal{E}_{MSMA}(\alpha,\beta) + \sum_{\mathbf{s}_i \notin \mathcal{S}_0} \mathcal{P}_T(\mathbf{s}_i)\mathcal{E}_{HSMA}(\alpha)\right], \qquad (2.5)$$

where $\mathcal{E}_{DSMA}$ is the energy dissipated by the **DSMA** block in detecting the input state, $\mathcal{E}_{MSMA}(\alpha,\beta)$ and $\mathcal{E}_{HSMA}(\alpha)$ is the energy dissipated by **MSMA** and **HSMA** block for each C_{opt} computation, $\sum_{\mathbf{s}_i \in \mathcal{S}_0} \mathcal{P}_T(\mathbf{s}_i)$ and $\sum_{\mathbf{s}_i \notin \mathcal{S}_0} \mathcal{P}_T(\mathbf{s}_i)$ is the probability that **MSMA** and **HSMA** block is used for C_{opt} computation, L is the number of samples after which the **SPA** block is reconfigured.

Substituting (2.4) and (2.5) in (2.3), we get the following optimization problem:

$$\min\ \mathcal{E}_{DAT}(\alpha,\beta)\ :\ s.t.\ 0 \le \alpha, \beta \le 1. \qquad (2.6)$$

Next, we apply this framework to frequency selective filters.

3. APPLICATION TO FREQUENCY SELECTIVE FIR FILTERING

For a given input signal $x(n)$ having stopband energy $E_{sb,x}$ dB, a filter is to be designed such that the stopband energy in filter output $y(n)$ is less than E_o dB. If $E_{sb,x} > E_o$, we need a frequency selective filter with stopband attenuation given by $A = (E_{sb,x} - E_o)$ dB. Hence, the input state is defined as $\mathbf{s} = E_{sb,x}$.

3.1. FIR Filter Design by Kaiser Window

We assume that an FIR filter $h(n)$ is to be designed such that the stopband attenuation is more than A dB and the transition bandwidth is less than $\Delta\omega$ radians. A windowed version $h(n) = w(n)h_{pr}(n)$ of the ideal prototype filter $h_{pr}(n)$ can be employed. A near-optimal window based on Bessel functions is given by Kaiser [5]. For this window, the tap length N [5] and the coefficient precision B_c [2] can be approximately obtained as

$$N = \frac{A}{2.285\Delta\omega}, \quad B_c = \frac{A}{6} + \frac{1}{2}\log_2\left(\frac{N}{12\epsilon}\right), \qquad (3.1)$$

where ϵ is a number much smaller than 1. For simplicity, we can assume $-0.5\log_2(12\epsilon)$ in (3.1) to be an integer.

3.2. SPA Block and Hardware Models

The **SPA** block for FIR filters is shown in Fig. 2. Let N_{max} be the total number of taps corresponding to the worst-case input state. Assume that the **HSPA** block has αN_{max} taps, which will provide a certain minimum level of stopband attenuation. The **RSPA** block will have $(1-\alpha)N_{max}$ taps, which can be dynamically reconfigured to achieve stopband attenuation $A(\mathbf{s})$ dependent upon input state $\mathbf{s} = E_{sb,x}$.

Assuming array multipliers, we enforce the following two reconfigurability modes for **RSPA** block:

1. Power-down : This is done by forcing the coefficient h_k to 0.

2. Variable-precision : This is achieved by shifting coefficients to the left such that the leading sign extension bits are removed. Thus the $B_c - \lceil \log_2 |h_k| \rceil$ rows of the array multiplier are powered down. To get the correct output, the product needs to be shifted back by $sh_k = B_c - \lceil \log_2 |h_k| \rceil$. Therefore, we employ barrel shifters at the output of each multiplier in Fig. 2.

Thus the **RSPA** configuration **C** is defined as,

$$C = [(h_1, sh_1), (h_2, sh_2), \cdots, (h_{(1-\alpha)N_{max}}, sh_{(1-\alpha)N_{max}})], \quad (3.2)$$

where h_k and sh_k are the inputs to the multiplier and the shifter in k^{th} tap of the **RSPA** block. The **HSPA** block is optimized off-line and is hardwired.

3.3. SMA Block

The **DSMA** block (Fig. 3) is employed to detect input state $\mathbf{s}(nL) = E_{sb,x}(nL)$ from a block of L input samples $x(nL), x(nL-1), \cdots, x(nL-L+1)$ as follows [3],

$$E_{sb,x}(n) = \rho \frac{1}{L} \sum_{i=0}^{L-1} [x^2(nL-i) - y^2(nL-i)], \quad (3.3)$$

where $y(n)$ is the output of the filter, and ρ is a constant dependent upon the stopband attenuation of the filter. For each state $\mathbf{s} = E_{sb,x}$, the **SMA** block determines minimum energy configuration $\mathbf{C}_{opt}(\mathbf{s})$ as defined in **Definition 3**. The βN_s optimum configurations $\mathbf{C}_{opt}(\mathbf{s}), \mathbf{s} \in \mathcal{S}_0$ are precomputed and stored in the **MSMA** block. The remaining $(1-\beta)N_s$ optimum configurations $\mathbf{C}_{opt}(\mathbf{s}), \mathbf{s} \notin \mathcal{S}_0$ are computed online by the **HSMA** block.

4. EXAMPLES

In the context of frequency selective filtering discussed in last section, we assume $E_o = -50$ dB, $\Delta\omega = 0.1\pi$. Assume further that an 8×8 programmable coefficient multiplier accumulate (MAC) in **RSPA** block dissipates 0.25 nJ per sample and an 8×8 constant coefficient and optimized MAC in **HSPA** block is assumed to dissipate 0.2 nJ per sample. We also assume that the energy dissipation per access of a $W_{mem} \times B_{mem}$ memory (W_{mem}=number of words, B_{mem}=number of bits per word) is given by [6],

$$\mathcal{E}_{mem} = \chi_0 + \chi_1 W_{mem} + \chi_2 B_{mem} + \chi_3 W_{mem} B_{mem}, \quad (4.1)$$

where $\chi_i, i = 0, 1, 2, 3$ are constants. Assuming 3.3V, 0.5μ technology, we employ typical values of $\chi_0 = 0.125$, $\chi_1 = 0.004$, $\chi_2 = 0.016$, $\chi_3 = 0.000125$ in nJ.

4.1. Input State Probability Function

We will assume that $\mathbf{s} = E_{sb,x}$ has the following probability mass function:

$$\mathcal{P}_S(s_i) = \kappa \exp\left[-\frac{1}{2}\left(\frac{s_i - \mu_s}{\sigma_s}\right)^2\right], \quad i = 1, 2, \cdots, N_s; \quad (4.2)$$

where μ_s is stopband energy corresponding to the nominal case and σ_s^2 indicates the spread of the stopband energy around the nominal value. Note that σ_s can be varied to generate different input distributions. For example, if $\sigma_s \to \infty$, $\mathcal{P}_S(\mathbf{s})$ reduces to uniform distribution and all the states become equally-likely. Similarly, if $\sigma_s \to 0$, $\mathcal{P}_S(\mathbf{s})$ reduces to a single state distribution with $\mathcal{P}_S(\mathbf{s} = \mu_s) = 1$. We assume $N_s = 32$, $\mathbf{s}_i = -50i/N_s$ and $L = 50$. Therefore, the input signal can have stopband energy $E_{sb,x} = \mathbf{s}$ from -50 dB to 0 dB.

4.2. Optimum Energy Solution

By employing (4.2), (2.2), (2.4) and (2.5), we formulate $\mathcal{E}_{DAT}(\alpha, \beta)$ defined in (2.3) for this application. We can then solve energy optimization problem in (2.6). Here, we vary the $\mathcal{P}_S(\mathbf{s})$ in (4.2) by sweeping σ_s and study its impact on the optimal solution.

Fig. 4 shows that $\mathcal{E}_{DAT,opt}$ increases with σ_s. This is because of the following two factors. First, as σ_s increases, the distribution varies from the nominal state to the uniform distribution thus increasing the probability of the worst state and hence $\mathcal{E}_{SPA,opt}$. Second, with increase in σ_s, the probability mass function gets distributed among more states thus increasing state transition probabilities and hence $\mathcal{E}_{SMA,opt}$.

Next, we investigate the energy savings $\mathcal{E}_{sav}$ as compared to the traditional hardwired design ($\alpha = 1$) corresponding to the worst case. The energy dissipation for this case is given by $\mathcal{E}_{SPA}(\alpha = 1)$. Fig. 5 shows that 35%-45% energy savings are achieved for given probability distributions.

As mentioned earlier, with increase in σ_s, the probability mass function gets distributed among more states thus requiring higher reconfigurability (lower α). This is consistent with the decreasing trend (with σ_s) of α_{opt}

(α is the fraction of the **SPA** block hardwired) in Fig. 6. Similarly, for higher σ_s (uniform distribution), β_{opt} increases (Fig.6) indicating that more configurations should be stored in the memory **MSMA**.

5. REFERENCES

[1] "Adaptive computing systems," *http://www.ito.darpa.mil/research/acs/index.html.*

[2] D. S. K. Chan and L. Rabiner, "Analysis of quantization errors in the direct form for finite impluse response digital filters," *IEEE Trans. Audio Electroacoustics*, vol. AU-21, pp. 354–366, Aug. 1973c.

[3] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing : An approach for energy efficient computing," in *Proceedings of International Symposium on Low Power Electronics and Design*, (Monterey, CA), pp. 347–352, Aug. 1996.

[4] M. Goel and N. R. Shanbhag, "Dynamic algorithm transformations (DAT) for low-power adaptive signal processing," *Proceedings of International Symposium on Low Power Electronics and Design*, pp. 161–166, Aug. 1997.

[5] J. F. Kaiser, "Nonrecursive digital filter design using the i_0-sinh window function adaptive," *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 20–23, Apr. 1974.

[6] P. Landman and J. M. Rabaey, "Activity-sensitive architectural power analysis," *IEEE Trans. Computer-Aided Design*, vol. 6, pp. 571–587, June 1996.

[7] K. K. Parhi, "Algorithm transformation techniques for concurrent processors," *Proceedings of the IEEE*, vol. 77, no. 12, pp. 1879–1895, Dec. 1989.

[8] J. Rabaey, "Reconfigurable processing : The solution to low-power programmable DSP," in *Proc. ICASSP*, (Munich, Germany), pp. 275–278, May 1997.

[9] N. R. Shanbhag and M. Goel, "Low-power adaptive filter architectures and their application to 51.84 Mb/s ATM-LAN," *IEEE Trans. Signal Processing*, vol. 45, no. 5, pp. 1276–1290, May 1997.

[10] E. Tsern and T. H. Meng, "A low-power videorate pyramid VQ decoder," *IEEE J. of Solid State Circuits*, vol. 31, no. 11, pp. 1789–1794, Nov. 1996.

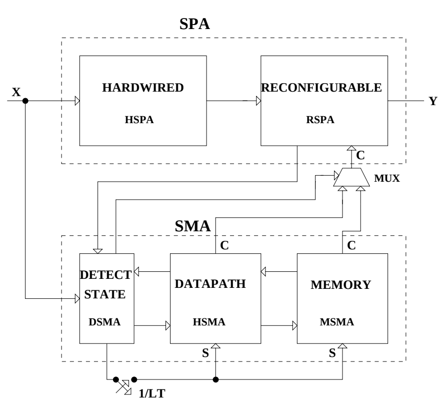

Figure 1: Block Diagram for DAT

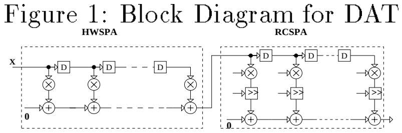

Figure 2: **SPA** Block for FIR filters

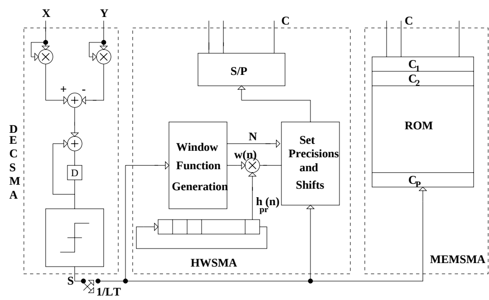

Figure 3: **SMA** Block for FIR filters

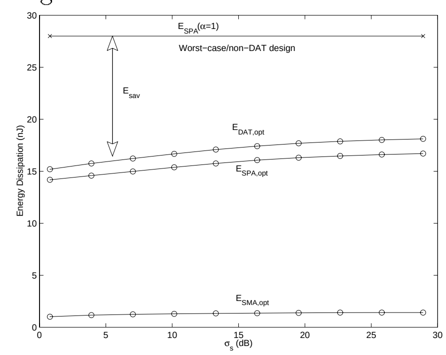

Figure 4: Optimal DAT Energy Dissipation versus σ_s

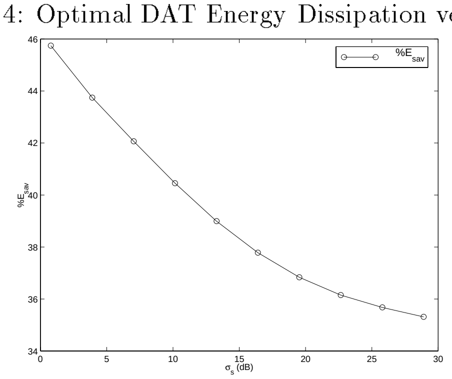

Figure 5: Energy Savings versus σ_s

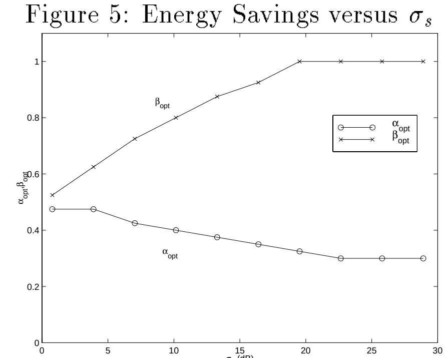

Figure 6: Optimal Partition α_{opt}, β_{opt} versus σ_s