

# Power Consumption of Parallel Spread Spectrum Correlator Architectures

Won Namgoong

Stanford University

Center for Integrated Systems 134

650-723-3941

won@now.stanford.edu

Teresa Meng

Stanford University

Center for Integrated Systems 209

650-723-3636

meng@mojave.stanford.edu

## 1. ABSTRACT

Parallel correlation in direct-sequence spread spectrum system allows faster and more reliable coarse acquisition. However, the power consumed becomes significant especially for receivers that employ a large number of parallel correlators. In this paper, the power efficiency of various parallel correlator architectures is explored assuming baseband sampled signals of two samples per chip. Active correlators placed in parallel that use both two's complement and sign-magnitude accumulators are first presented. A functionally equivalent M-parallel passive correlators are then studied. In this approach, the baseband sampled signals are passed through a tapped delay-line. Each tap is then multiplied by a stationary reference pseudonoise code and summed using a binary tree network. The passive correlators are generally more power efficient compared to both types of active correlators, especially for large M values. Further reduction in power consumption is possible by splitting the tapped delay-line into even and odd delays and summing using two smaller binary tree adders. This proposed architecture consumes significantly less power compared to all other architectures. The power dissipation of M-parallel correlator architectures are evaluated for  $M = 8, 16, 32$  using TSMC 0.35- $\mu\text{m}$  CMOS technology at 3.3V supply voltage.

## 2. INTRODUCTION

The first step in a direct-sequence spread spectrum (DS/SS) receiver is coarse acquisition. This is the process of successively correlating the local despreading code with the incoming spreading pseudonoise (PN) code until the two codes are in alignment within a half code-chip interval. Coarse acquisition typically consists of a two-dimensional search in time (code phase) and frequency. Because this process can be quite time-consuming, many modern receivers perform a parallel search with M correlators. The M parallel correlators are also employed to achieve more

reliable coarse acquisition in channels with multipaths. Furthermore, for proper RAKE reception, detecting signal strengths of different multipaths are necessary for proper combining, and this is usually achieved by using M parallel correlators. The M correlators operate at a common frequency with a half code-chip offset among each of them.

In portable communication applications, minimizing power consumption is important for extending the battery life of a portable unit. Correlators typically consume a large fraction of the overall receiver power [1]. One reason for the large power consumption in the correlators is that they usually operate at the highest frequency in a DS/SS receiver. Each correlator accumulates the reference PN modulated signal for a certain amount of time, which is typically the period of a data bit, then dumps its sum for further processing. Since correlation usually occurs at twice the chip frequency and the output of the correlator is decimated at the data frequency, the correlators operate at a high frequency in a DS/SS receiver. The power consumed by the correlators become especially significant for receivers that employ a large number of parallel correlators.

This work compares the power consumption of four different parallel correlator architectures for  $M = 8, 16, 32$ . Circuit layouts of each architecture and M value were generated using Cascade's Epoch silicon compiler. Epic's PowerMill was then employed to estimate the power consumption of these circuit layouts for TSMC 0.35- $\mu\text{m}$  CMOS technology at 3.3V supply voltage. We assume input signals to the correlators are obtained by sampling at exactly two samples per chip. Furthermore, the baseband samples are assumed uncorrelated with all of the M-parallel correlators.

## 3. ACTIVE PARALLEL CORRELATORS

The active M-parallel correlators are illustrated in figure 1. The  $i$ th baseband signal, denoted as  $b(i)$ , is correlated with M active correlators, labelled correlator0 to correlator(M-1). Each active correlator consists of a multiplier, an accumulator, and a tristate buffer. For the  $j$ th active correlator,  $b(i)$  is multiplied by a single bit reference PN code  $x(i-j)$ . Since baseband signals consist of two samples/chip, the value of each reference PN code is duplicated, such that,  $x(0) = x(1)$ ,  $x(2) = x(3)$ , etc. Furthermore, the indices of reference PN codes of adjacent correlators differ by one, because adjacent correlators are separated by half a code-chip offsets.

Multiplication of  $b(i)$  with  $x(i-j)$  in the  $j$ th correlator is achieved by performing a XOR operation. If  $x(i-j) = 0$ ,  $b(i)$  is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
ISLPED98, Monterey, CA, USA  
© 1998 ACM 1-58113-059-7/98/0008..\$5.00

negated and a value of one is added to the accumulator since numbers are represented in two's complement. After correlating for  $2N$  samples, where  $N$  is the desired number of chips correlated, the result is dumped to the output bus for further processing.

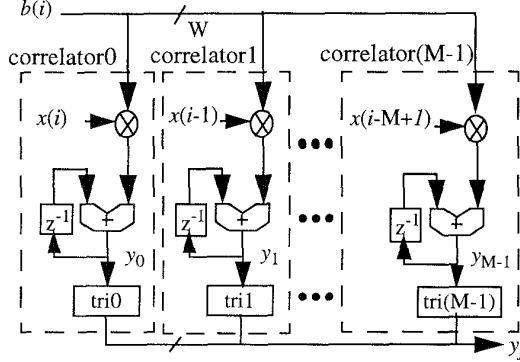


Figure 1. M-parallel active correlator architecture

Before initiating the correlation process, the accumulators are sequentially reset from the accumulator in the 0th to the (M-1)st correlator. After correlating  $2N$  samples, the correlation results drive the bus by sequentially enabling the 0th tristate to the (M-1)st tristate. The output of the  $j$ th correlator is

$$y_j = \sum_{i=j}^{2N+j-1} b(j)x(i-j). \quad (1)$$

If  $b(i)$  is  $W$  bits, the accumulator in figure 1 is  $W + \log_2(2N)$  bits wide. Since XOR operation and accumulation are performed for every sample, an appropriate  $W + \log_2(2N)$  bit adder type that meets the timing requirement must be selected. We assume that the clock frequency,  $f_{sample}$ , is low enough that a ripple adder is sufficient. A ripple adder is the slowest compared to other adder types, but it is the most power efficient. Hence, the power consumption values presented in section 5 for the active parallel correlators represent a lower bound.

Since  $b(i)$  and  $x(i-j)$  are uncorrelated by assumption, all the input bits to the accumulator are toggling with probability 0.5. To reduce the amount of toggling, the accumulation can be achieved using a sign-magnitude number representation instead of a two's complement number representation [1]. The active correlator power consumption is then reduced by approximately 30% [2]. Assuming a reduction of exactly 30% in the correlators, the power dissipation of  $M$  active parallel correlators with sign-magnitude accumulators are compared to other parallel correlators in section 5.

## 4. PASSIVE PARALLEL CORRELATORS

### 4.1 Correlator with binary tree adder

$M$ -parallel passive correlators are functionally equivalent to a  $M$ -parallel active correlators described in the previous

section. In a  $M$ -parallel passive correlator architecture, the  $i$ th baseband signal  $b(i)$  is passed through a tapped delay line consisting of  $M-1$  registers as shown in figure 2. Each delayed  $b(i)$  values are multiplied by reference PN codes that are indexed as shown in the figure. The multiplied values, which are implemented by XOR's, are combined by an adder network. The resulting sum,  $z(i)$ , is integrated with the sum  $M$  cycles earlier, which is stored in a FIFO. This integration with the sum  $M$  cycles earlier is performed  $2N/M$  times to achieve the effect of correlating  $N$  chips. Mathematically,

$$z(i) = \sum_{l=0}^{M-1} b(i-l)x\left(\left\lfloor \frac{i-(M-1)}{M} \right\rfloor M + M - 1 - l\right). \quad (2)$$

After integrating  $z(i)$   $2N/M$  times with the sum  $M$  cycles earlier, the resulting value is the output of the  $j$ th active correlator given in (1). That is,

$$y_j = \sum_{i=j}^{2N+j-1} b(j)x(i-j) = \sum_{k=1}^{2N/M} z(kM+j-1). \quad (3)$$

The correlation results are available sequentially from  $y_0$  to  $y_{M-1}$ . Thus, the results after the final integration are dumped sequentially in the same order for further processing as in the  $M$ -parallel active correlator architecture.

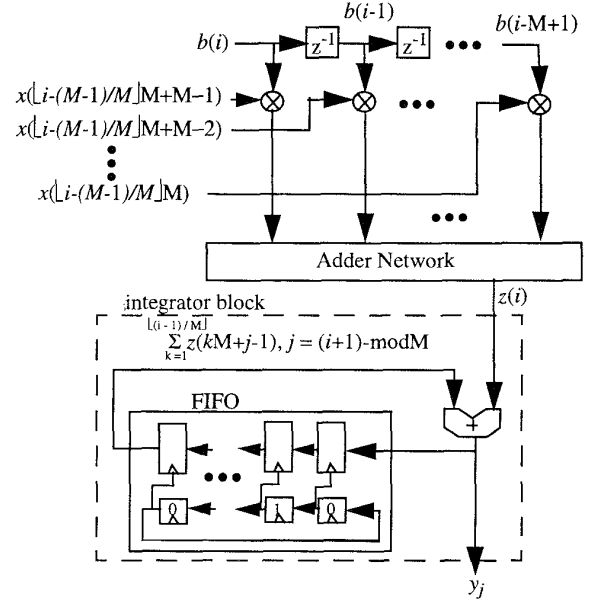


Figure 2. Passive correlator with binary tree architecture

In figure 2, the adder network for the  $M$  multiplication results is implemented as a binary tree with 4:2 adders and a vector merger. Assuming the inputs to the correlators are  $W$ -bits and  $M$  is some power of two, the binary tree requires a total of  $\log_2(M)-1$  stages. A vector merger is necessary since

the result of the binary tree summation is in CSA form. To reduce the delay through in the adder network, registers are placed before and after the vector merger. The vector merger is a  $(W + \log_2(M))$ -bit ripple adder.

The integrator block in figure 2 consists of a FIFO and an integrator. The integrator is a  $W + \log_2(2N)$  bit carry lookahead adder. The FIFO consists of  $M$  stages each  $W + \log_2(2N)$  bits wide. It is designed to minimize switching activity by employing a pointer such that only one register experiences clock and output transistions [3]. The power consumed by this FIFO, however, is significantly larger than that of a single register, because a global bus is connected to each input and output, increasing both the input and the output loads.

#### 4.2 Correlator with splitting binary tree adder

Whenever  $b(i)$  is shifted in figure 2, half of the addition performed by the adder network is redundant since  $x(0) = x(1)$ ,  $x(2) = x(3)$ , etc. All odd delays of  $b(i)$  are multiplied and summed by the same reference PN code as in the previous cycle. Thus, even delays of  $b(i)$  (e.g.  $b(i), b(i-2), \dots$ ) are providing new correlation information; whereas, odd delays of  $b(i)$  (e.g.  $b(i-1), b(i-3), \dots$ ) are not providing any new correlation information. In order to exploit this redundancy, an architecture is proposed in figure 3. The odd and even delays of  $b(i)$  are split into two branches, allowing

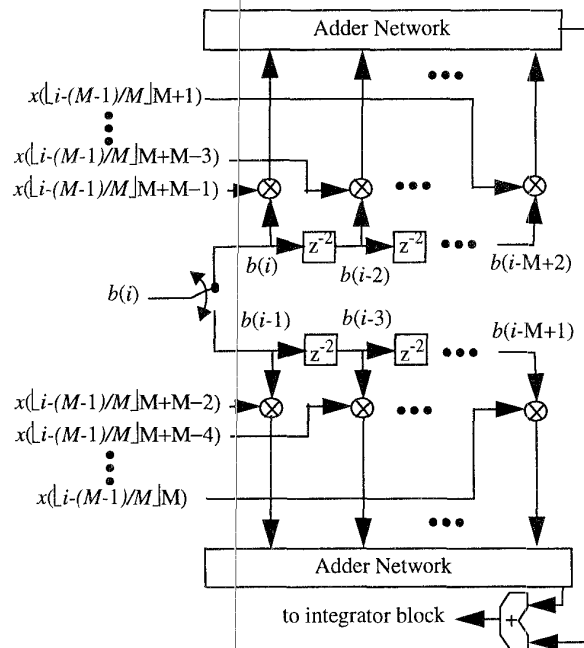


Figure 3. Passive correlator with split binary tree adder

each to operate at  $f_{sample}/2$ . By alternatively shifting the samples in even and odd delay branches, only new correlation information is processed. Each adder is a binary tree network that requires summation of almost half as many values. The summation result is integrated as in figure

2 using an integrator block, which is not shown in figure 3.

## 5. RESULTS AND ANALYSIS

Assuming  $W = 4$ , normalized power consumption is plotted against  $M$  when  $N = 1024$  in figure 4 for four different parallel correlator architectures: active correlator using two's complement accumulator, active correlator using sign magnitude accumulator, passive correlator with binary tree adder, and passive correlator with split binary tree adder. We refer to these parallel correlator architectures as architectures A, B, C, and D, respectively. Architecture D is the most power efficient, followed by C, B, then A. The relative power consumption among the various architectures becomes especially distinctive for large  $M$  values. This is because as  $M$  increases, the contribution to the total power is dominated by the addition operation, that is, the accumulators for the active correlators (architectures A and B) and the adder networks for the passive correlators (architectures C and D). The accumulators are less efficient at summing a stream of  $M$  numbers, requiring significantly more full adders than the tree adder network. Thus, architecture C consumes less power compared to architectures A and B. Since architecture D is a modified architecture C that exploits the duplication of the PN reference codes, the former outperforms the latter. Architecture A can also be altered to take advantage of the duplication of the PN reference codes, but based on figure 4, it would still perform worse than architecture C. The proposed architecture D is the most power efficient architecture, because it employs a tree adder to efficiently sum a stream of  $M$  numbers and exploits the redundancy due to duplication of the PN reference codes to minimize total power consumption.

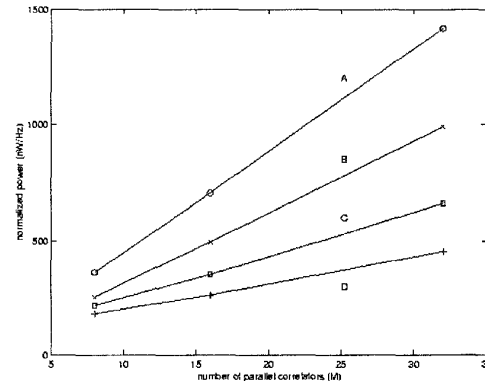


Figure 4. Normalized power consumption vs.  $M$

## 6. REFERENCES

- [1] Sheng, S. et al., "A low-power CMOS chipset for spread spectrum communications" 1996 Proceedings of the IEEE ISSCC, p 346-347
- [2] A. Chandrakasan, Low Power Digital CMOS Design, Ph.D. Thesis, UC Berkeley, Berkeley, CA, 1994
- [3] E. Tsern et al., "A low power video-rate pyramid VQ decoder" IEEE JSSC, November 1996