AREA COMPLEXITY OF MULTILECTIVE MERGING

Pavel Feriánč
Ondrej Šýkora

Computing Centre, Slovak Academy of Sciences, Dúbravská 9
842 35 Bratislava, Czecho-Slovakia

Abstract. Lower bounds on the area $A(n,m,k,r)$ required for merging of two sorted sequences of $k$-bit numbers with length $n$ and $m$ respectively, when the inputs can be replicated up to $r$ times ($r \leq n$), are given:

\[
A(n, m, k, r) = \begin{cases} 
\Omega\left(\frac{n}{r}\right) & \text{for } 2^k \geq \frac{n}{r} \text{ and } n \geq m \geq \frac{n}{r} \\
\Omega\left(m\left(\left\lceil \log \frac{n}{rm}\right\rceil + 1\right)\right) & \text{for } 2^{\frac{\alpha k}{r}} \geq \frac{n}{r} \text{ and } \frac{n}{r} \geq m \\
\Omega\left(m\left(\left\lceil \log \frac{2^k}{m}\right\rceil + 1\right)\right) & \text{for } \frac{n}{r} \geq m \text{ and } \frac{n}{r} \geq 2^{\frac{\alpha k}{r}} \geq m \\
\end{cases}
\]

where $K > 0$ is the constant

INTRODUCTION

The paper analyzes the following problem: let $m,n \in \mathbb{N}$ and $m \leq n$, let $D_n = \{ (x_1, \ldots, x_n) | x_1 \leq \ldots \leq x_n; x_i (i = 1, \ldots, n) \text{ consist of } k \text{ bits } \}$ and let $D_m = \{ (z_1, \ldots, z_m) | z_1 \leq \ldots \leq z_m; z_i (i = 1, \ldots, m) \text{ consist of } k \text{ bits } \}$. The merging problem can be characterized by the function $f : (x_1, \ldots, x_n, z_1, \ldots, z_m) = (y_1, \ldots, y_{n+m})$, where $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}, Z = \{z_1, \ldots, z_m\}$ and $Y = \{y_1, \ldots, y_{n+m}\}$ satisfying either $y_i \in X$ or $y_i \in Z$. Without loss of generality let us assume that $m, n, r, k$ be powers of 2.

The memory of our circuit consists of square units of area (each one with area $\lambda^2$ ($\lambda > 0$)) and at most one bit can be stored per unit of area. The i/o schedule is assumed to be when- and where-determinate. Determinate schedules, which require prespecified times and locations for the input and output of each bit, are discussed in [U]. Tight lower bounds on the area required for merging with semitective inputs (each data is read once) were shown in [PSV]. Our area bounds are proved for r-multilective input; i.e. data can be read more than once but at most r-times. All computations and temporary storage of data, however, must be done within the merging device. According to [Si] we can assume w.l.o.g. that in a time unit $t$ at most one input bit is supplied or at most one output bit is delivered and only one input or output event occurs.
The motivation for the study of the multilective circuits is that it leads either towards more general techniques for searching area bounds or it brings more general results. It is evident that multilectivity enables substantially diminished area of circuit. For example, in the work [G], a language with the following property is described. The area required for its recognition is $\Omega(\sqrt{n})$, but if allow each input to be read twice, then the area for its recognition is $O(1)$.

According to our knowledge until now there were proven only two results about nontrivial area lower bounds for concrete multilective problems. The area $A(n, k, r) = \Omega(\log n + 2^k \log \frac{n}{2^{k+1} - 1})$ for $2^k \leq \frac{n}{r}$; $A(n, k, r) = \Omega(\log n + \frac{n}{r} \log \frac{r^2 + 1}{n})$ for $\frac{n}{r} \leq 2^k \leq n^{O(1)}$ for sorting $n$ $k$-bit numbers is showed in [Si] and the area $A = \Omega(\frac{n^2}{r})$ for the matrix product of $n \times n$ matrices is shown in [Sa].

Now we give a brief introduction to the technique of our proofs (a similar technique was used in [Si]). A time interval $\tau$ of the entire time of computation is chosen. Call the inputs (outputs) which are read (delivered to output) during the interval $\tau$ $\tau$-inputs ($\tau$-outputs) and the inputs (outputs) for which no copy is read during the interval $\tau$ as non-$\tau$-inputs (non-$\tau$-outputs). Set the $\tau$-inputs in a proper way so that the $\tau$-outputs are dependent on the non-$\tau$-inputs. The number of all various vectors of $\tau$-outputs is equal to the number of all various vectors of corresponding non-$\tau$-inputs, which is also the number of possible circuit states. The lower bound on the required area is the logarithm of the number of the circuit states.

**LOWER BOUNDS**

**Theorem 1.** If $2^k \geq \frac{n}{r}$ and $n \geq m \geq \frac{n}{r}$, then $A(n, m, k, r) = \Omega(\frac{n}{r})$.

To prove this theorem we need the next lemma.

**Lemma.** Let $b_1...b_t$, where $t$ is even, be a string of zeros and ones such that the number of zeros is equal to the number of ones. Then arbitrary $s \leq t$ satisfies: there exists $j$ ($0 \leq j \leq t - s$) such that substring $b_{j+1}...b_{j+s}$ contains at least $\lfloor \frac{1}{3}s \rfloor$ ones.

**Proof of Lemma:** Three cases are possible:

1. $\lfloor \frac{3}{4}t \rfloor \geq s > \frac{1}{2}t$  
   If there are at least $\lfloor \frac{1}{3}s \rfloor$ ones in the substring $b_{s+1}...b_t$ then $j = t - s$. Otherwise the substring $b_1...b_s$ must contain at least $\frac{t}{2} - (\lfloor \frac{1}{3}s \rfloor - 1) > \lfloor \frac{1}{3}s \rfloor$ ones.

2. $t \geq s > \lfloor \frac{3}{4}t \rfloor$  
   Let $t = s + i$. Hence $i < \lfloor \frac{1}{3}s \rfloor$. Let us assume the substring $b_{s+1}...b_t$ contains only ones (it means $i$ ones). Then the string $b_1...b_s$ must contain at least $\frac{s+i}{2} - i > \lfloor \frac{1}{3}s \rfloor$ ones.

3. $s \leq \frac{1}{2}t$  
   Let us divide the string $b_1...b_t$ to three substrings so that two substrings of them have the length $s$ and the last one has the length $i$. Let in each of these substrings be at most $(\lfloor \frac{1}{3}s \rfloor - 1)$ ones i.e. $3(\lfloor \frac{1}{3}s \rfloor - 1)$ ones together. However, in the whole string $b_1...b_t$ there must be $2s+i$ ones. Since $3(\lfloor \frac{1}{3}s \rfloor - 1) < \frac{2s+i}{2}$ there exists one substring, which contains at least $\lfloor \frac{1}{3}s \rfloor$ ones.