Communication costs of Schönhage-Strassen fast integer multiplication

UC Berkeley ParLab
Derrick Coetzee, Jim Demmel, Oded Schwartz

Derrick Coetzee   Jim Demmel   Oded Schwartz

Main results

- Tight lower and upper bounds on the communication cost of Schönhage-Strassen integer multiplication:

$$\mathrm{IO}(n) = \Theta\left(n \frac{\log n}{\log M} \log \frac{\log n}{\log M}\right)$$

n is the length of the input numbers
M is the fast memory size

Schönhage-Strassen

- FFT-based multiplication algorithm
- Multiplies two n-bit integers with $O(n \log n \log\log n)$ arithmetic complexity
  - Compare long multiplication, $O(n^2)$
- Asymptotically fastest practical algorithm

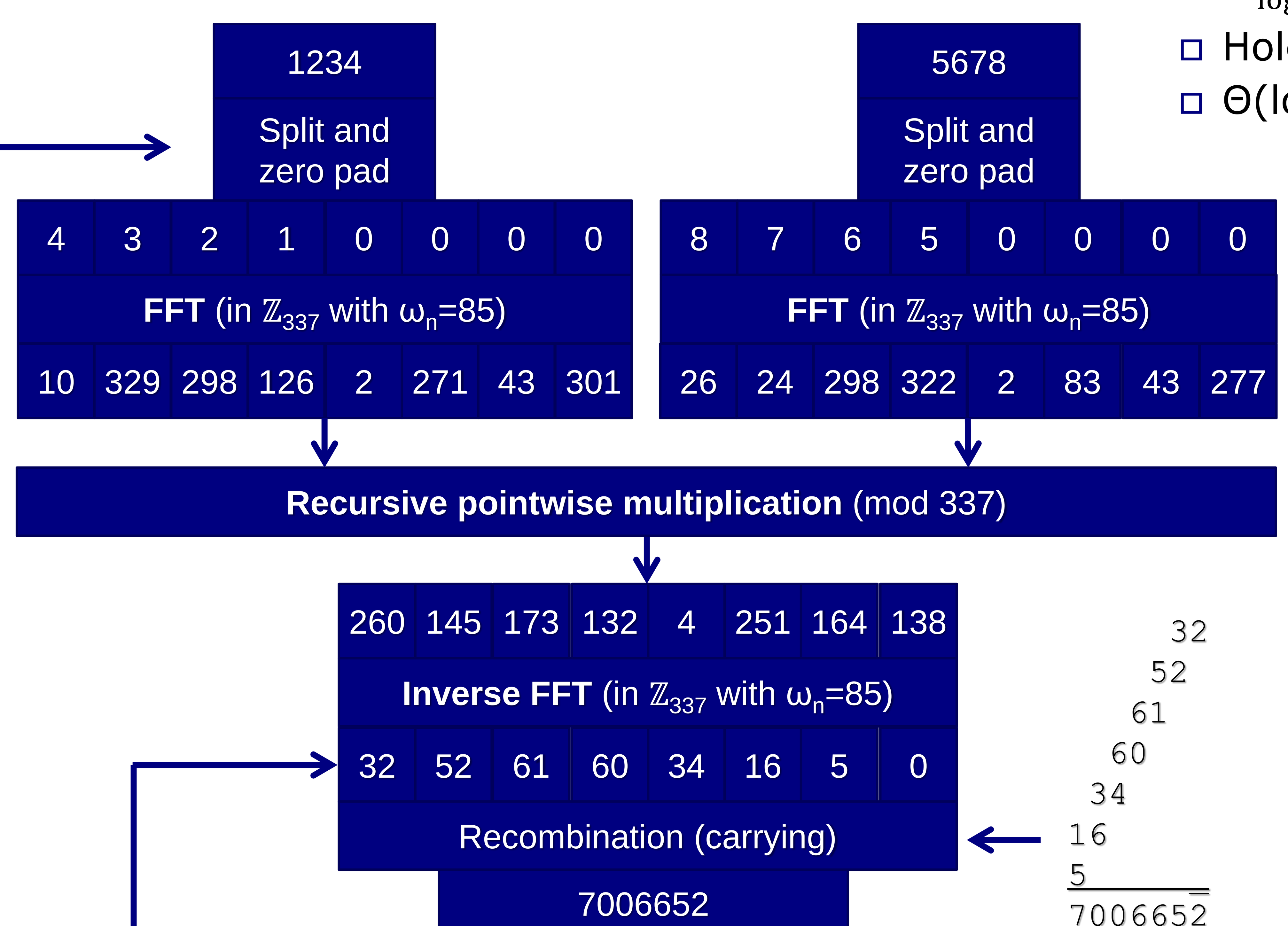
The algorithm

Split inputs into d-length vectors of m-bit numbers ($dm = n$)

- Note: Can eliminate zero-padding using negacyclic convolution

- Compute acyclic convolution:
  - FFT, pointwise multiplication, IFFT
- Do all operations in ring $\mathbb{Z}_{2^{n'}+1}$
  - Multiplications during FFT become shifts/adds
  - Modular reductions after recursive multiplications become shifts/adds

```
        1  2  3  4
        5  6  7  8
        8 16 24 32
     7 14 21 28
  6 12 18 24
5 10 15 20
5 16 34 60 61 52 32
```




```
      32
     52
    61
   60
  34
 16
5
7006652
```

Arithmetic complexity

- Largest partial sum requires $2m + \lg d$ bits
- Optimal d: $O(\sqrt{n})$

```
               11    11
               11    11
             1001  1001
     1001    1001
     1001   10010  1001
```

$d(2^m-1)^2$

- $T(n) = \sqrt{n}\, T(2\sqrt{n}+½\lg n) + \Theta(n \log n)$
  - $\Theta(n \log n)$ total work per level
  - $\Theta(\log\log n)$ levels
- $T(n) = O(n \log n \log\log n)$

IO - complexity

Communication cost of **FFT**:

- IO = $\Omega(n \log n/\log M)$ [Hong & Kung 1981, Savage 1995]
- IO = $O(n \log n/\log M)$ [Frigo, Leierson, Prokop, Ramachandran 1999]

Communication cost of Integer multiplication:

Phased implementations:
Each FFT done independently

- $\mathrm{IO}(n) = \begin{cases} \sqrt{n}\cdot \mathrm{IO}(2\sqrt{n}) + \Theta\left(\frac{n \log n}{\log M}\right) & \text{if } n > 3M \\ \Theta(M) & \text{otherwise} \end{cases}$
- $\Rightarrow \mathrm{IO}(n) = \Theta\left(n \frac{\log n}{\log M} \log \frac{\log n}{\log M}\right)$

Interleaved implementations cannot do better:
Proof: Impose reads/writes before/after each FFT

- $\mathrm{IO}(n) \geq \begin{cases} \sqrt{n}\cdot \mathrm{IO}(2\sqrt{n}) + c\frac{n \log n}{\log M} - 2n & \text{if } n > 3M \\ \Theta(M) & \text{otherwise} \end{cases}$
- $c\frac{n \log n}{\log M} - 2n > c' \frac{n \log n}{\log M}$ for $n > M^{2/(c-c')}$
- Holds for all but O(1) recursion levels
- $\Theta(\log\log n)$ levels do $\Theta(n \log n / \log M)$ I/O each.

Future Work

- Apply to other multiplication algorithms: [Strassen 1968],[Knuth 1997],[Fürer 2007],[De, Saha, Kurur, Saptharishi 2008]
- Extend to other multiplication algorithms: [Karatsuba 1962], [Toom 1963],[Cook 1966]
- Hybrids of the above
- Apply to polynomials multiplication
- Implement, predict, and test performance
- CA-Parallel algorithms