Sieving Hardware for the NFS: Architectures and their Bottlenecks

Willi Geiselmann
The Number Field Sieve

- Precomputation
- Relation collection
- Linear Algebra (Matrix step)
- Postprocessing
Relation Collection

- Given $F_1(x,y)$, $F_2(x,y) \in \mathbb{Z}[x,y]$, homogeneous polynomials, e.g. of degree 5 and 1
- Find $(a,b) \in \mathbb{Z} \times \mathbb{N}$ with $F_1(a,b)$ and $F_2(a,b)$ smooth, $\gcd(a,b) = 1$
Parameters for 1024 Bit
(from TWIRL, 2003)

- Smoothness bounds:
  \( B_1 = 2.6 \cdot 10^{10} \) (algebraic),
  \( B_2 = 3.5 \cdot 10^9 \) (rational). \( \approx 2 \times 5 \text{ GByte} \)

- Sieving region:
  \( A = 5.5 \cdot 10^{14}, \ -A < a < A; \)
  \( B = 2.7 \cdot 10^8, \ 0 < b < B. \) \( \approx 1400 \text{ TByte} \)
Suggested Hardware Designs

- **TWINKLE** [Shamir 1999; Shamir, Lenstra 2000]  
  not designed for 1024 bit numbers

- **TWIRL** [Shamir, Tromer 2003]  
  full wafer design

- **Mesh-based sieving** [G., St. 2003, 2004]  
  not feasible for 1024 bit numbers

- **SHARK** [Franke et al. 2005]  
  elaborated butterfly transport system

- **SmallChips** (Non-Wafer-Scale Sieving HW) [G., St. 2007]  
  more realistic, but high inter-chip communication
## Sieving (TWINKLE/TWIRL)

<table>
<thead>
<tr>
<th>Counter</th>
<th>37</th>
<th>31</th>
<th>29</th>
<th>23</th>
<th>19</th>
<th>17</th>
<th>13</th>
<th>11</th>
<th>7</th>
<th>5</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><img src="image1.png" alt="Points" /></td>
<td><img src="image2.png" alt="Points" /></td>
<td><img src="image3.png" alt="Points" /></td>
<td><img src="image4.png" alt="Points" /></td>
<td><img src="image5.png" alt="Points" /></td>
<td><img src="image6.png" alt="Points" /></td>
<td><img src="image7.png" alt="Points" /></td>
<td><img src="image8.png" alt="Points" /></td>
<td><img src="image9.png" alt="Points" /></td>
<td><img src="image10.png" alt="Points" /></td>
<td><img src="image11.png" alt="Points" /></td>
<td><img src="image12.png" alt="Points" /></td>
</tr>
</tbody>
</table>

The table above illustrates the sieving process for various counters. Each counter has a corresponding time index, indicating the progress in the sieving process.

**Time:**

- 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

**Summation:**

- \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\) \(\sum\)
TWIRL - Types of Primes

- Largish primes (rational): \(2^{19} < p\)
  (algebraic): \(2^{22} < p\)
- Medium primes: \(256 < p\)
- Smallish primes: \(p < 256\)
TWIRL - Largish Stations

DRAM

- Rotates / reads with constant speed
- Writes to the "correct" address
TWIRL - Largish Stations
Adder Array

- **Rational:**
  - 2,100 lines
  - 4,096 columns
  - $\approx \frac{2^{23}}{2}$ adders

- **Algebraic:**
  - 14,900 lines
  - 32,768 columns
  - $\approx \frac{2^{29}}{64}$ adders
TWIRL - Med./Small Stations

- For $p < 2^{19}$ ($2^{22}$)
- Stations are different: smaller DRAM,
  more logic to generate "multiple hits"
- Adder Array similar ($\approx 500$ lines)
TWIRL - Parts/Communication (rational)

- Size: 160 cm²
  (60 cm² DRAM)

- L. Stations:
  60 cm² (incl. DRAM)

- Adder Array:
  64 cm² + 35 cm²

(when using a 0.13 µm process)
TWIRL - Parts/Communication (algebraic)

- Size: 659 cm²
  (435 cm² DRAM)
- L. Stations:
  490 cm² (incl. DRAM)
- Adder Array:
  130 cm² + 39 cm²
  (when using a 0.13 µm process)
TWIRL - Performance

- Total chip area $8 \times 160 \text{ cm}^2 + 659 \text{ cm}^2$
- One sieving line in 33 sec (1 GHz)
- Sieving of a 1024 bit number with 194 devices in one year
- Fastest design (time x area)
TWIRL – Problems/Solutions

- Devices cannot be cut into pieces (I/O bandwidth of chips).
- Larger Factorbasis increases this problem.
- Production errors especially in Adder Array cause problems: Redundancy required (increases size).
- Smaller production process and/or significant increase in I/O bandwidth would help.
SmallChips - Idea

- Sieve intervals of a given size ($2^{26}$)
- Generate (the rare) hits of large primes on different chips
- Collect the hits in the memory cell responsible for this sieve location
SmallChips - Types of Primes

- Largish primes I: $2^{27.2} < p < B_1 < 2^{35}$
  ...Type II/III: $1.5 \cdot 10^7 < p < 2^{27.2}$
- Medium primes: $2^{13} < p < 1.5 \cdot 10^7$
- Smallish primes: $p < 2^{13}$
SmallChips - Largish Stations

- A ≤ r < -A + S
DRAM for (p, r)-pairs

- A + S ≤ r < -A + 2S
DRAM for (p, r)-pairs

control logic & adder

control logic & adder
SmallChips - Largish Stations

- 256 stations for $p > 1.5 \cdot 10^7 \approx 2^{27.2}$
- Distributed on 32 chips:
  - size: 472 mm$^2$ (0.13 µm process)
  - output: 448 bit per clock cycle
  - memory: 99%, logic: 1%
- DRAM to store both FBs: 160 cm$^2$
SmallChips - Medium/Smallish Stations

Different type of storage:

- First \((p,r)\)-pair are stored, others are calculated
- For \(p < 2^{20}\): calculated in the collection unit (reduces communication, increases storage/area)
SmallChips - Collection Unit

- Distributed on 4 chips, each holding 4 arrays of 32 x 32 counting units.
- Each unit is in charge of $2^{12}$ sieve locations,
- and adding up the $\log(p)$ values.
SmallChips - Collection Unit (area estimates)

Distributed on 4 chips:
size: 493 mm²
(0.13 µm process)
input: 3584 bit / cc
memory: 94%
logic: 6%
SmallChips - Performance

- Total silicon area 172 cm\(^2\)
- One subinterval \(S=2^{26}\) in 53,000 cc
- One sieve line in 25 min (600 MHz)
- Sieving of a 1024 bit number with 8300 devices in one year
- 3.5 x more silicon area than TWIRL
- or 2.0 x more after modification
SmallChips - Problems/ Solutions

- Very fast communication as input to collection unit -> distribute collection unit on more chips.
- Smaller process reduces chip size and/or allows to increase FB, communication will not increase much.

\[
\begin{align*}
4\% \ FB & \quad \leftrightarrow \quad 0.4\% \ communication \\
100\% \ FB & \quad \leftrightarrow \quad 10\% \ communication
\end{align*}
\]
Conclusion

- SmallChips seems to be feasible
- Design/production costs are high
- Running costs are very high: 8300 devices require 1.6 MWatt (200 W per device seems optimistic)
- -> 1 400 000 € per Factorization