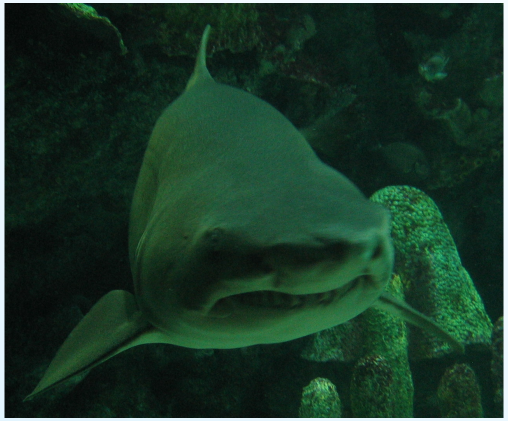
SHARK

A Realizable Special Hardware Sieving Device for Factoring 1024-bit Integers

Jens Franke, Thorsten Kleinjung - University of Bonn
Christof Paar, Jan Pelzl - University of Bochum
Christine Priplata, Colin Stahlke - EDIZONE GmbH, Bonn

Outline

- Why SHARK - Factoring 1024-bit Integers?
- General Number Field Sieve and Lattice Sieving
- SHARK Sieving Device - Architectural Overview
- Butterfly Transport System
- Cost Estimates
- Conclusions



Factoring 1024-bit Numbers

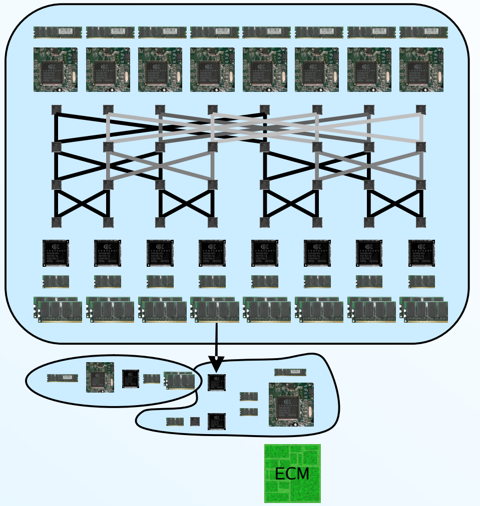
- seemed impossible some 20 years ago
- seems possible in some (near?) future with very large ASICs for some million dollars
- seems possible today with conventional computers for some thousand million dollars

Can we do it with today´s conventional technology for less than thousand million US dollars?

SHARK

SHARK uses lattice sieving to perform the sieving step of GNFS for a 1024-bit integer within a year for less than 200 million US dollars.

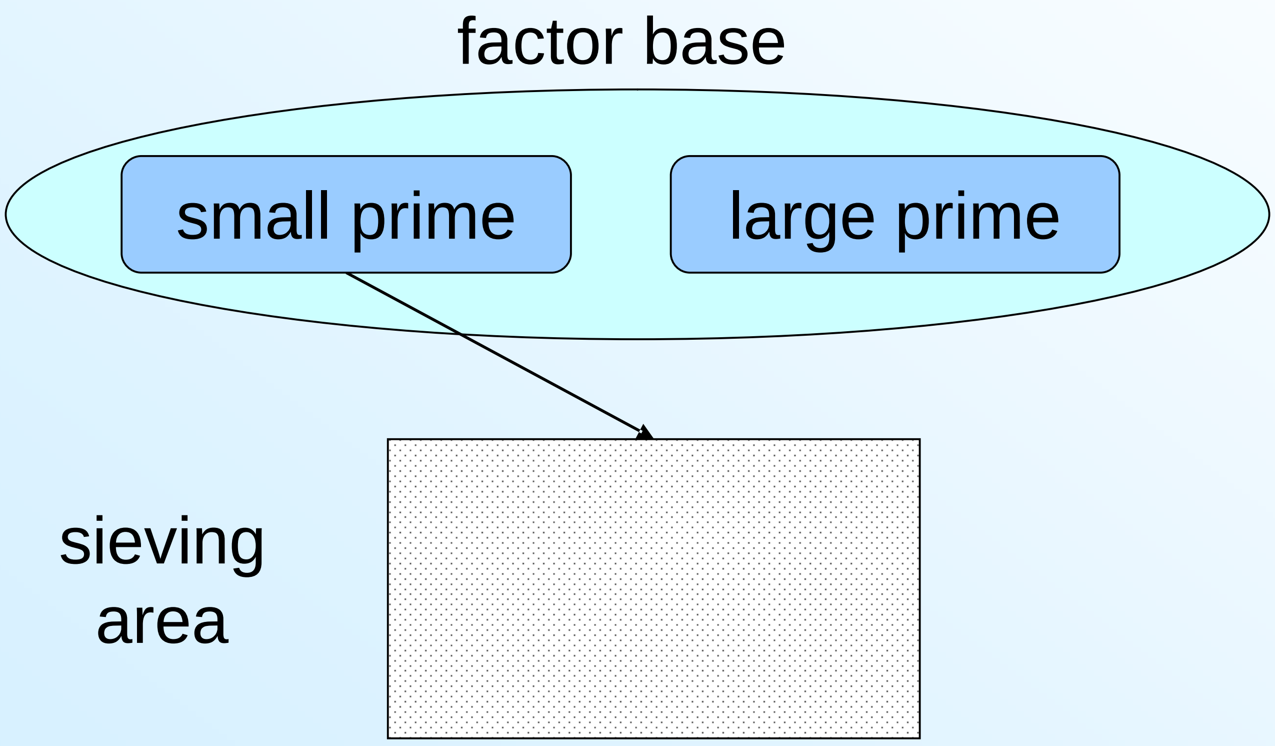
- 2300 identical machines
- small specialized ASICs
- of-the-shelf RAM
- modular architecture
- conventional data buses



The price (without development costs) is an upper bound and can be lowered considerably by changing the parameters.

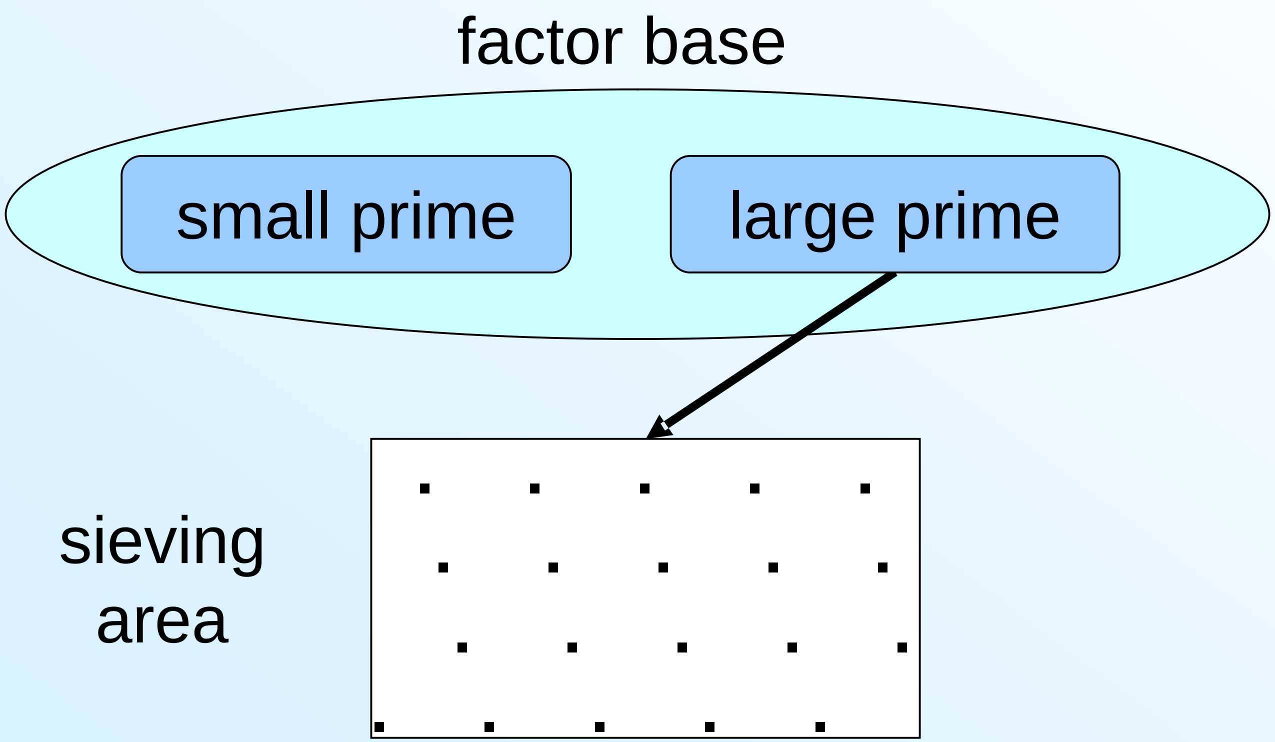
General Number Field Sieve

The GNFS is asymptotically the best algorithm to factor RSA moduli.



General **N**umber **F**ield **S**ieve

The GNFS is asymptotically the best algorithm to factor RSA moduli.

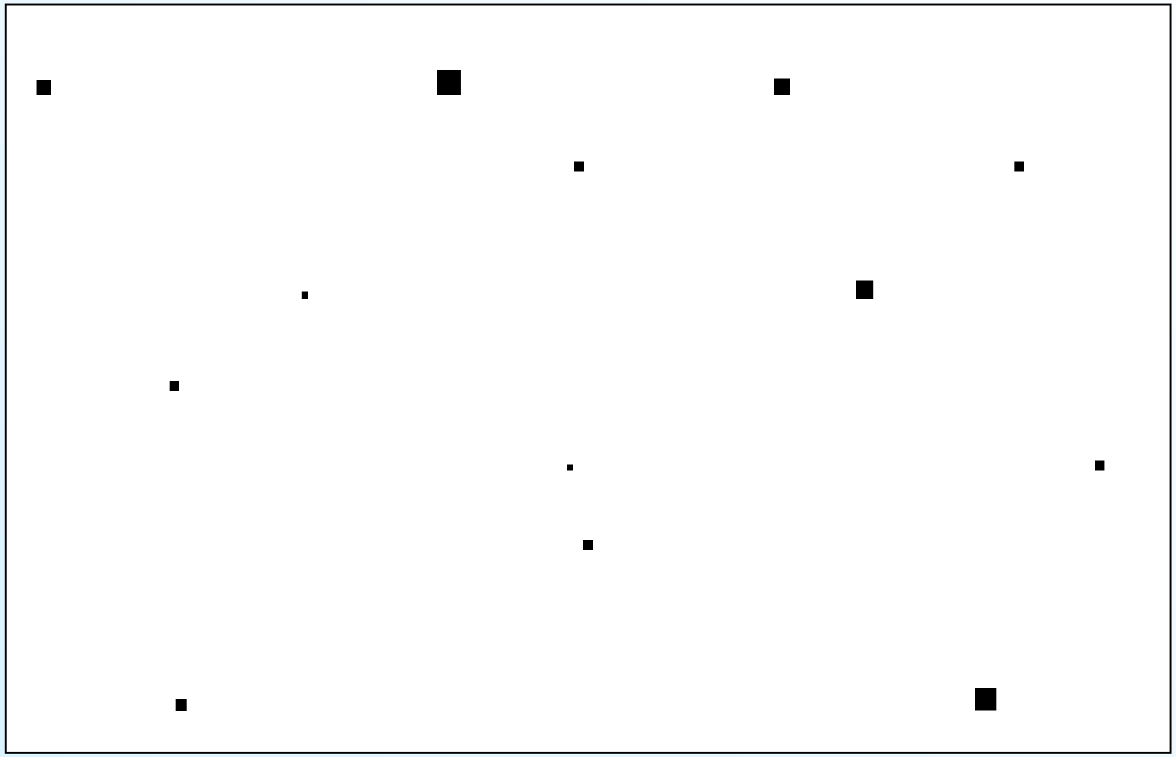


General **N**umber **F**ield **S**ieve

The GNFS is asymptotically the best algorithm to factor RSA moduli.

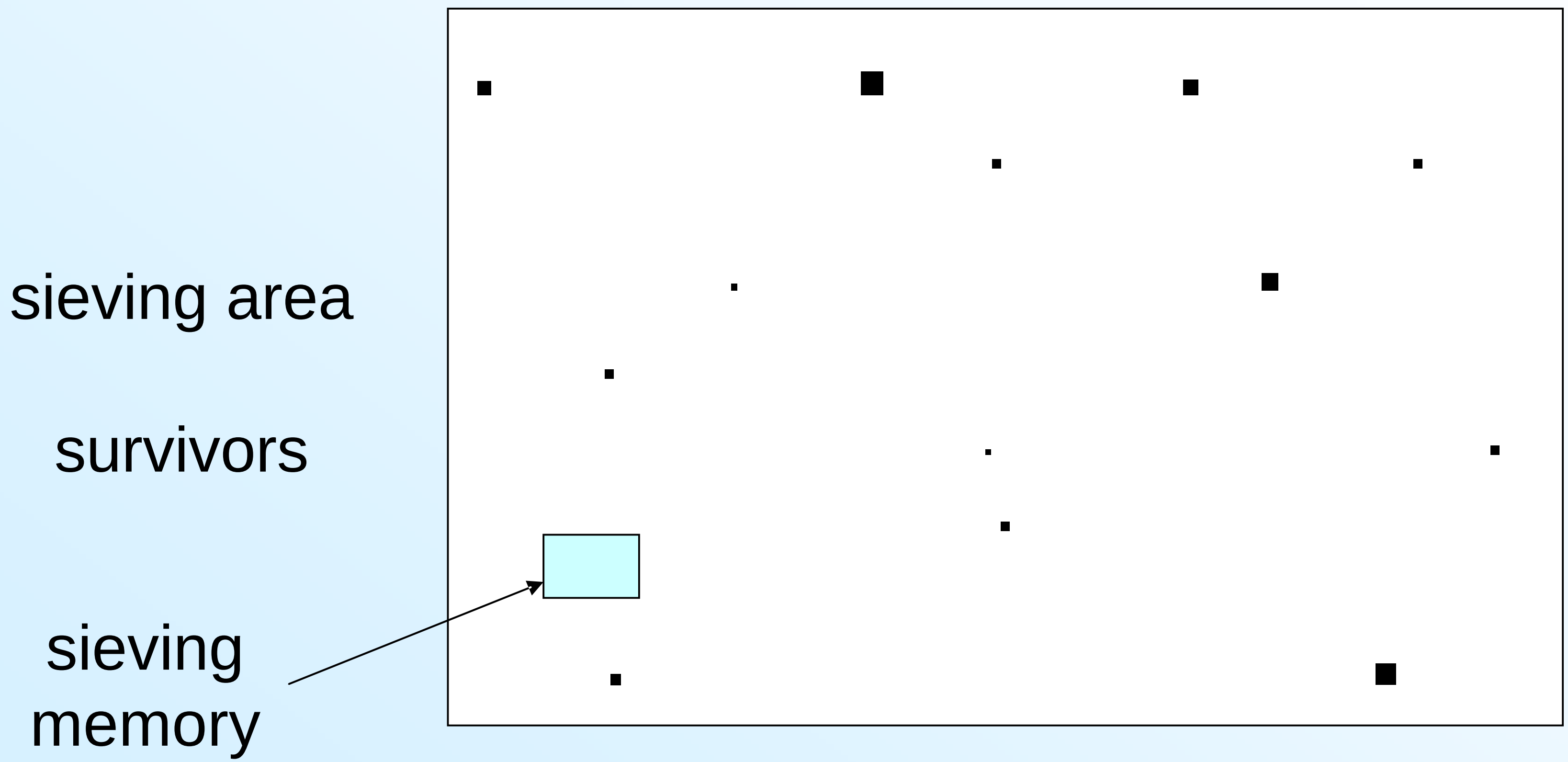
sieving area

survivors

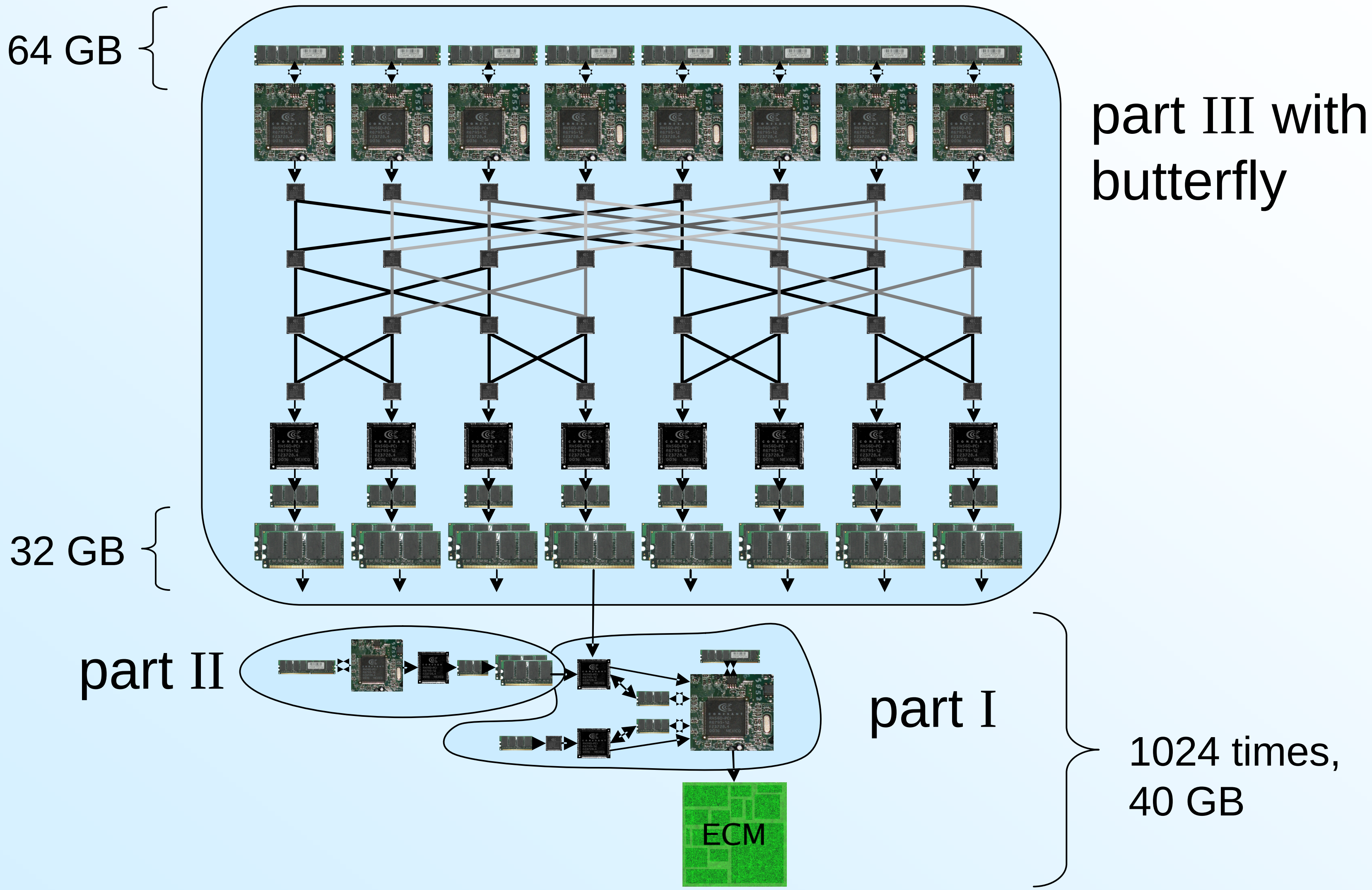


General **N**umber **F**ield **S**ieve

The GNFS is asymptotically the best algorithm to factor RSA moduli.



SHARK Architecture

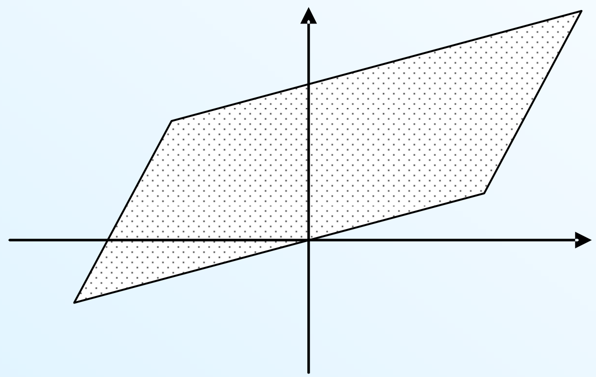


Factor Base and Sieving Area

factor base F = { (p,r) } p prime, r integer, ...

sieving area **A** = { (a,b) } a,b coprime integers and some conditions:

- for line sieving: $|a|<A,\ 0<b<B$
- for lattice sieving: size conditions on (a,b), (q,s) "special q" $q \mid a+b\cdot s$



Sieving Procedure

- Create triples (p, log p, e) such that p contributes at e:

  $(p,r) \in F, \quad e = (a,b) \in \mathbf{A}, \quad p \mid a+b\cdot r$

- "Sort" triples w.r.t. 3rd entry

- For each position e in the sieving area **A** check if

$$\sum_{(p,\log p,e)} \log p \quad > \quad \text{bound depending on e}$$

Divide et impera!

Facilitate the sorting of triples:

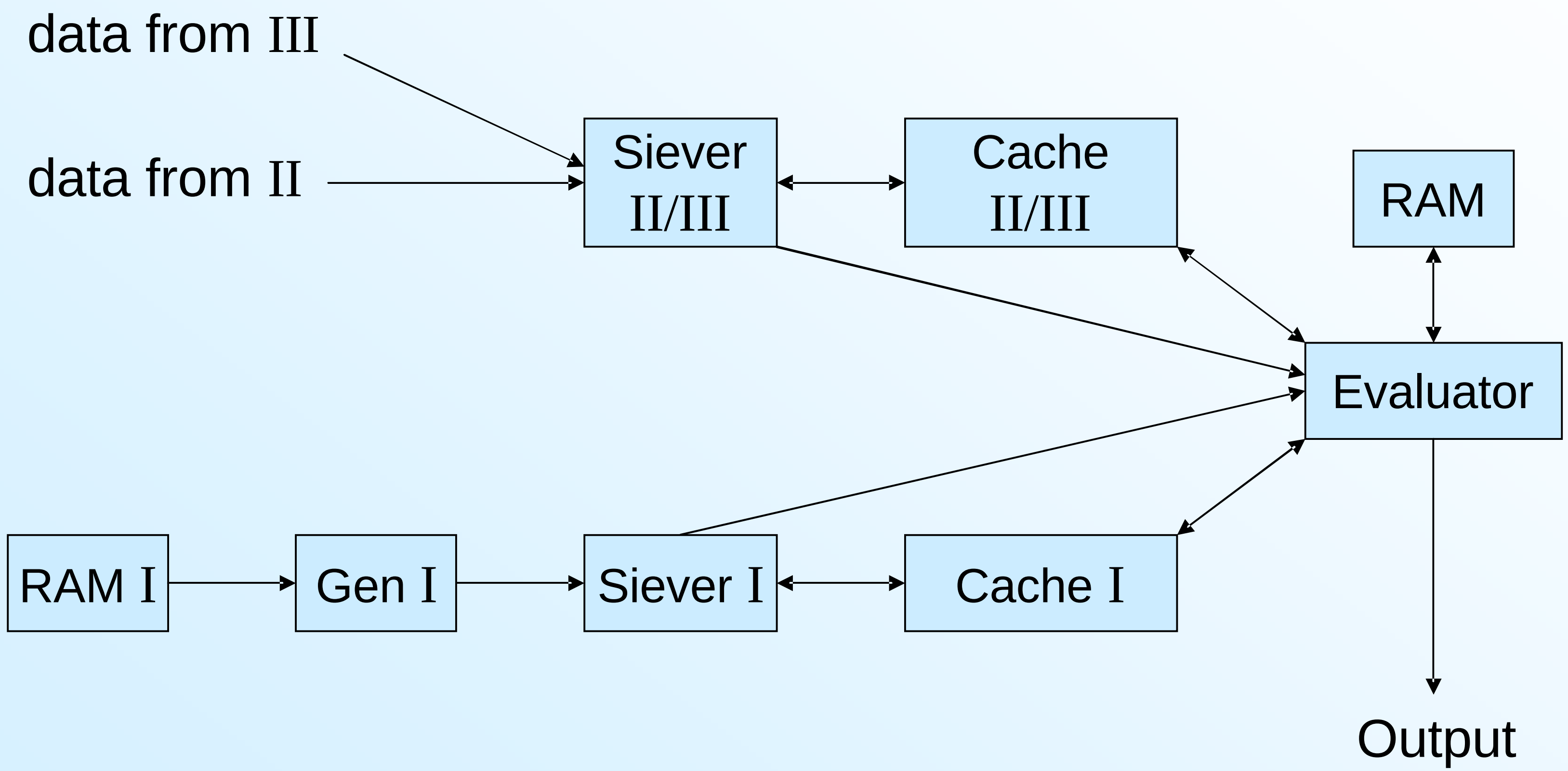
- Subdivide the sieving area **A** in chunks of size 2^{14}.
- Subdivide the factor base F in types I, II and III:

$$\text{I}: \quad p < 2^{14}$$

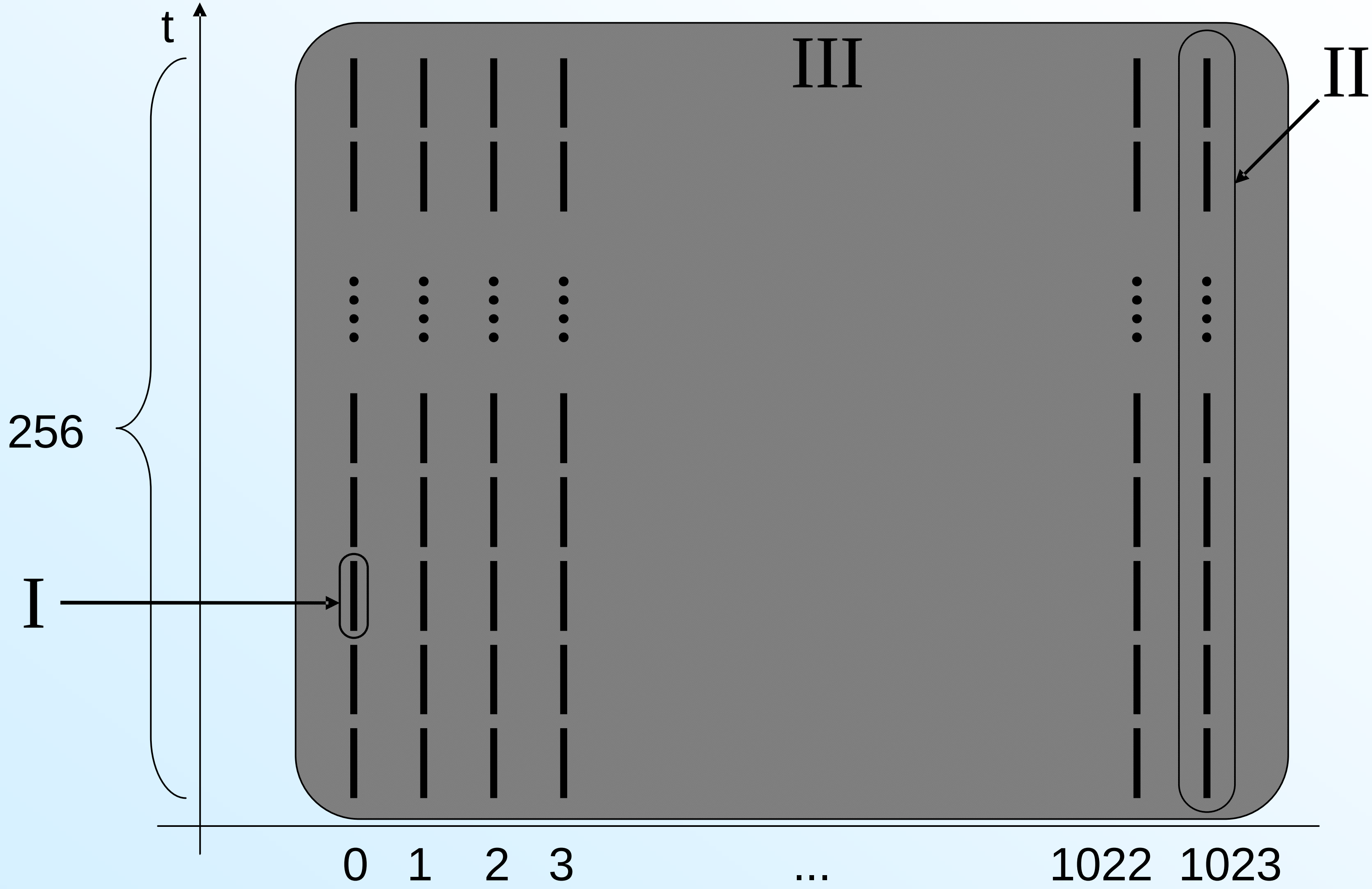
$$\text{II}: \quad 2^{14} < p < 2^{22}$$

$$\text{III}: \quad 2^{22} < p$$

Part I


data from III
data from II
Siever
II/III
Cache
II/III
RAM
Evaluator
RAM I
Gen I
Siever I
Cache I
Output

Partition of the Sieving Area



Generation of Triples (p, log p, e)

I on the fly

II for 256 chunks, local, store in 256 arrays

III for 2^{18} chunks, global

$\Longrightarrow$ need to transport triples to destination,

then same procedure as in II

Triples of Part III

- Part III of the factor base is distributed among the 1024 parts of the machine.

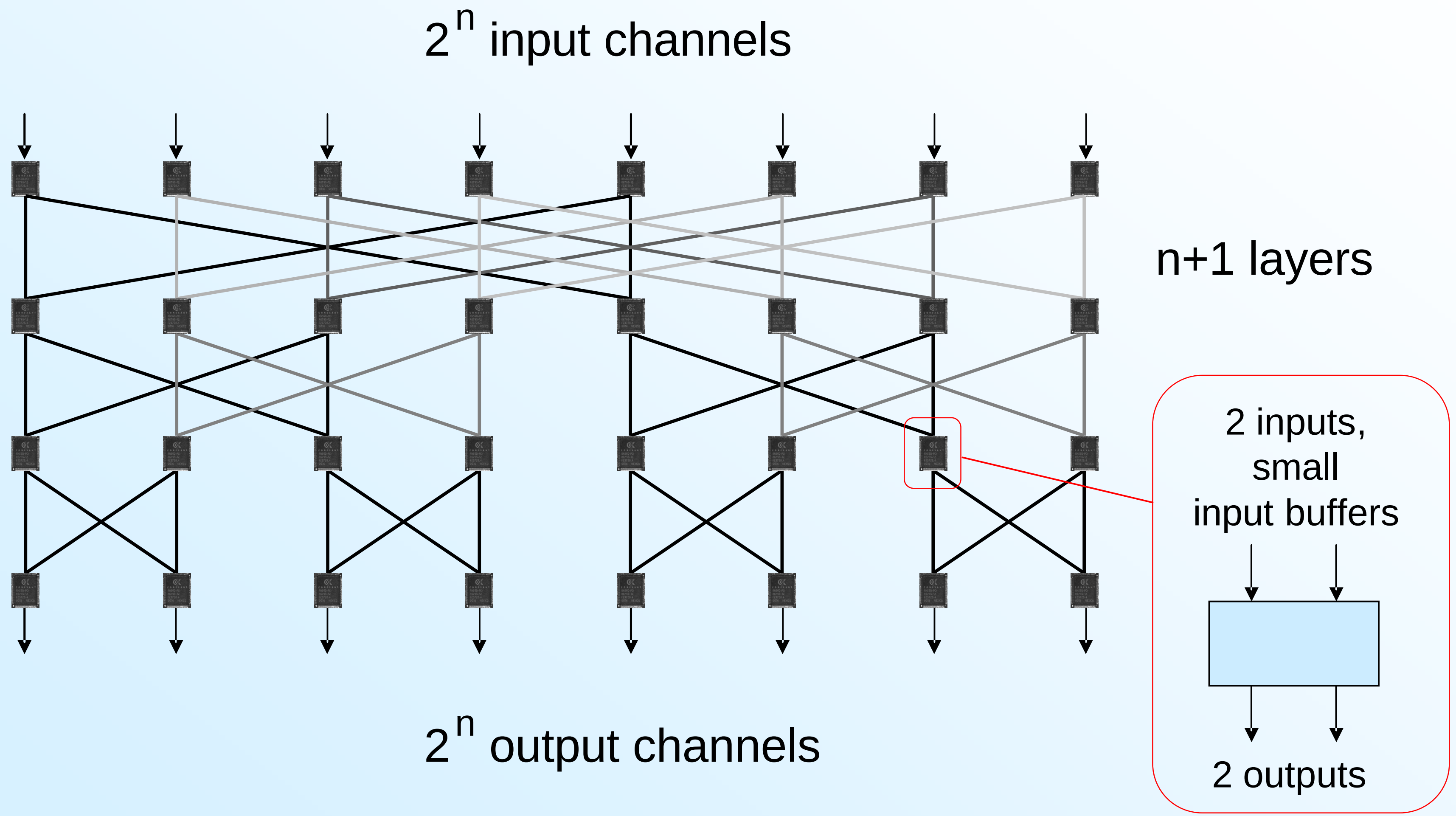
- In each step generate triples for an area of size 2^{32}.

- Send the triples via a transport system to the destination.

Details about the generation of these triples follow in the next talk "Continued Fractions and Lattice Sieving".

Butterfly Transport System

SHARK: n=10



Butterfly Transport System

• exactly one path from each input to each output

• path at each junction depending on one bit of e

Options for the realization in hardware:

• Layers can be permuted.

• Several junctions can be realized as one device or ASIC.

Doubling the Width of the Transport System

- decreases the runtime by a factor of 2
- doubles the number of processors outside the transport system
- total memory remains constant, but individual memory chips become smaller
- transport system is duplicated and gets an additional layer (long connections!)

Rough Cost Estimates

1 machine, ASIC:

memory:	136 GB RAM + 192 MB cache	21 000 $
processors:	1/4 wafer + transport system	9 000 $
power supply + additional electronic + cooling:		30 000 $
PCs (control) + ECM (negligible):		10 000 $
		70 000 $
power consumption: 30 kW	per year	25 000 $

2300 machines complete the sieving step in one year and cost

160 million US $ + 60 million US $ electricity.

Conclusions

SHARK can perform the sieving step for a 1024-bit integer factorization in 1 year and costs around 200 million US $ (pessimistic estimate).

- small ASICs and conventional memory chips
- possible improvements:
  - larger transport system
  - better choice of parameters
  - more ECM (this afternoon)
- realizable with today´s technology

Any questions?