SHARCS2007
2007/9/9
Wiena Mariott

CAIRN 3:An FPGA Implementation of the Sieving Step with the Lattice Sieving

FUJITSU LTD.
Tetsuya Izu, Jun Kogure, *Takeshi Shimoyama

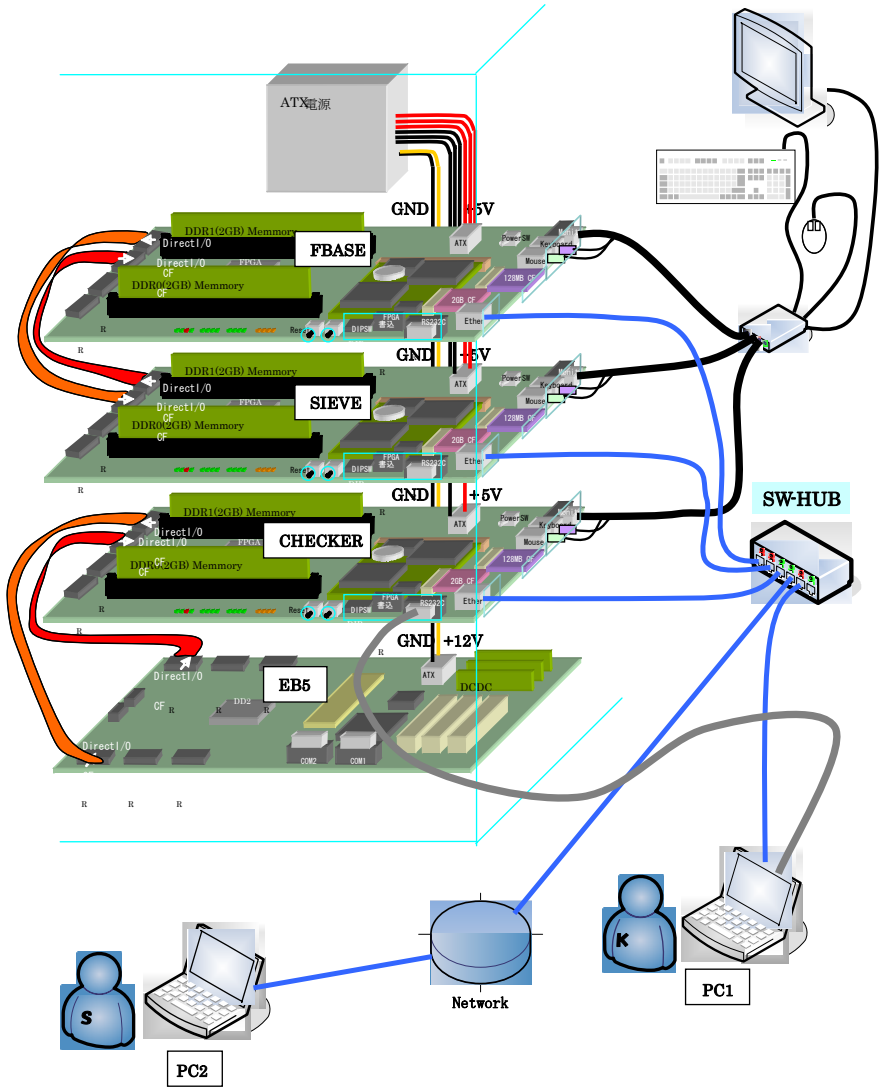
A part of this research is financially supported by a contract research with the National Institute of Information and Communications Technology (NICT), Japan

What is CAIRN3

FUJITSU

Circuit Aided Integrated Relation Navigator

CAIRN 3
Special Purpose Hardware
for Lattice Sieving



Outline of This Talk

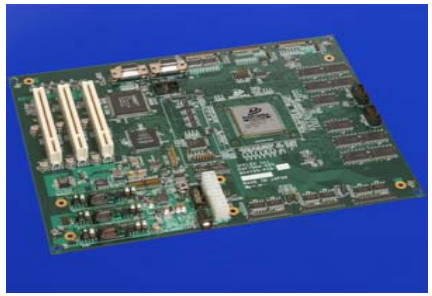
FUJITSU

1. Summary of the Integer Factoring by using a Hardware
2. Problems on the Implementation of the Lattice Sieving and Our Solutions
3. Security Evaluation of the RSA

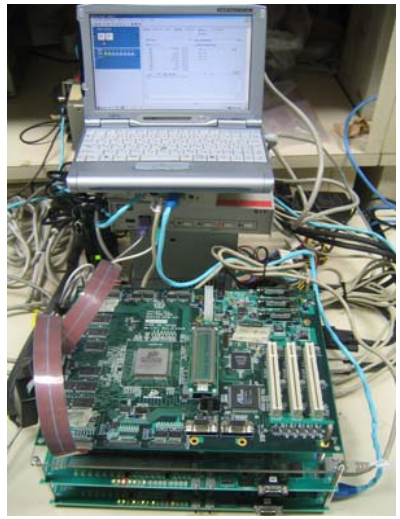
Progress of the Development of the Sieving HW

FUJITSU

- In 2005 (CAIRN1) [SHARCE2005]
  - Line Sieving
  - Implemented on DAPDNA2
- In 2006 (CAIRN2) [CHES2007]
  - Line Sieving and Relation checking
  - Implemented on FPGA × 2 and DAPDNA2
- In 2007 (CAIRN3) [This presentation]
  - Lattice Sieveing and Relation Checking
  - Implemented on FPGA × 3 and DAPDNA2



CAIRN1

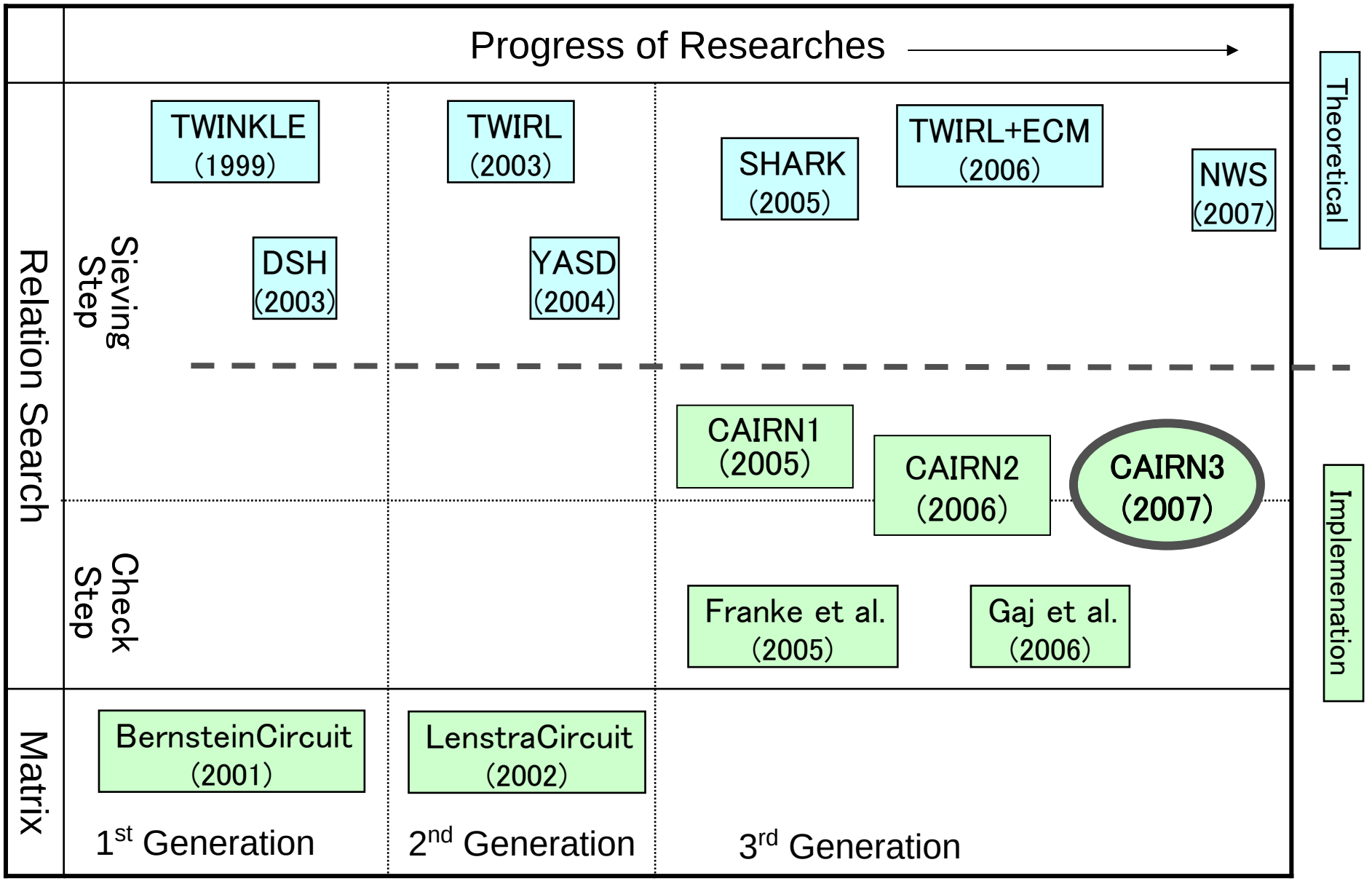


CAIRN2

CAIRN3

Previous Works on the Integer Factoring HW

FUJITSU

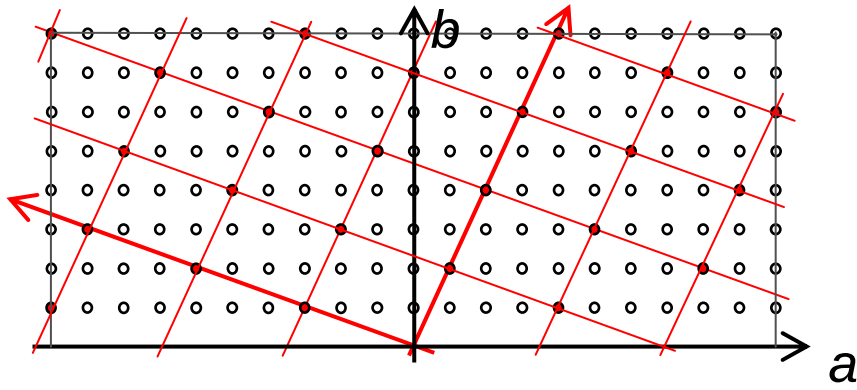


Lattice Sieving Method

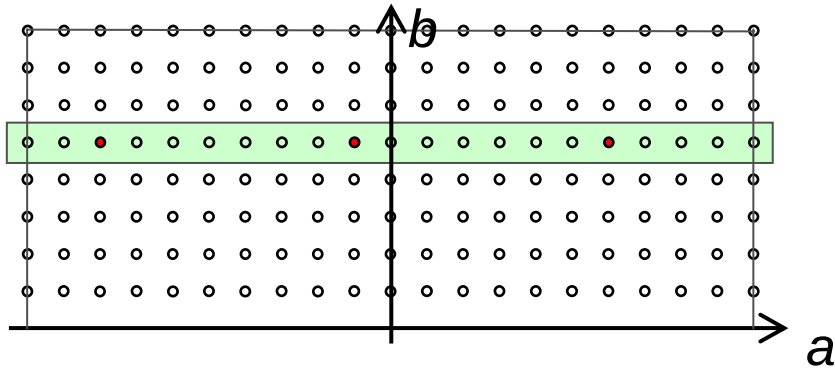
FUJITSU

The Lattice Sieving Method is

- One of the procedure for the Sieving step in NFS
- Proposed by Pollard in 1991
- Using “Lattice Space” generated by Large Factor Base
- Used for Achiving the current Integer Factoring Records by using NFS
- The details of the implemenation are written in few documents



Lattice Sieve



Linear Sieve

Problems for the Lattice Sieving

FUJITSU

There are 3-large problems for the Lattice Sieving.

1. Computation of the Lattice Base

   Need large amount of the multiplications and divisions

2. Management of the Lattice Base

   Most of all Lattice base (99%) is pass through the circuit without any operations if an easygoing way had been used.

3. Making the round of the Lattice Points

   Complicated Judgement of the Boundary of the Sieving Space

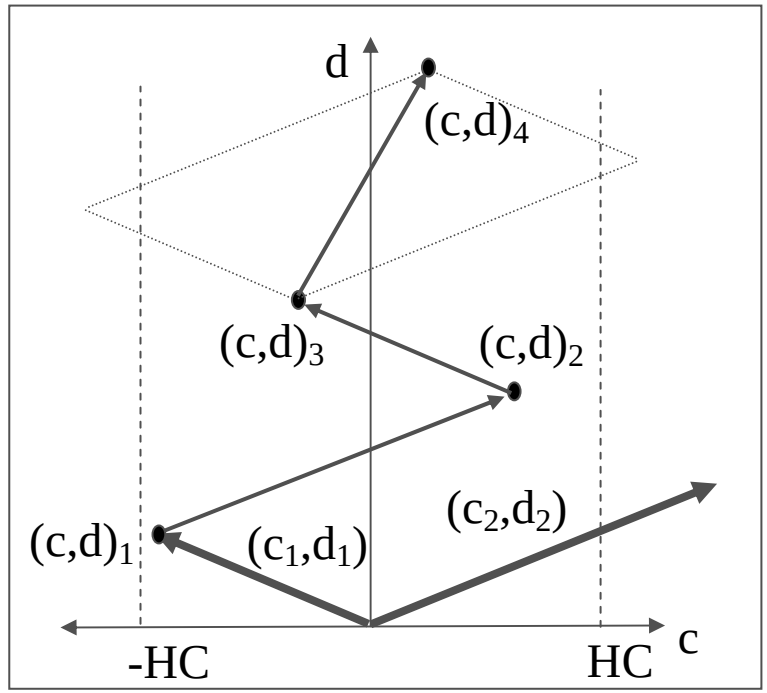
3. How to round the Lattice Points (1)

FUJITSU

Lattice Sieving developed by Franke & Kleinjung(※)

In the case that the lattice base satisfies “some conditions”, the all of the lattice points in the sieving space are treated by the following equation.

$$(c',d') = (c,d) + \begin{cases} (c_1,d_1) & c + c_1 \geq -HC \\ (c_2,d_2) & c + c_2 < HC \\ (c_1,d_1) + (c_2,d_2) & \text{otherwise} \end{cases}$$



※ Franke, Kleinjung, “Continued Fractions and Lattice Sieving”, SHARCS2005.

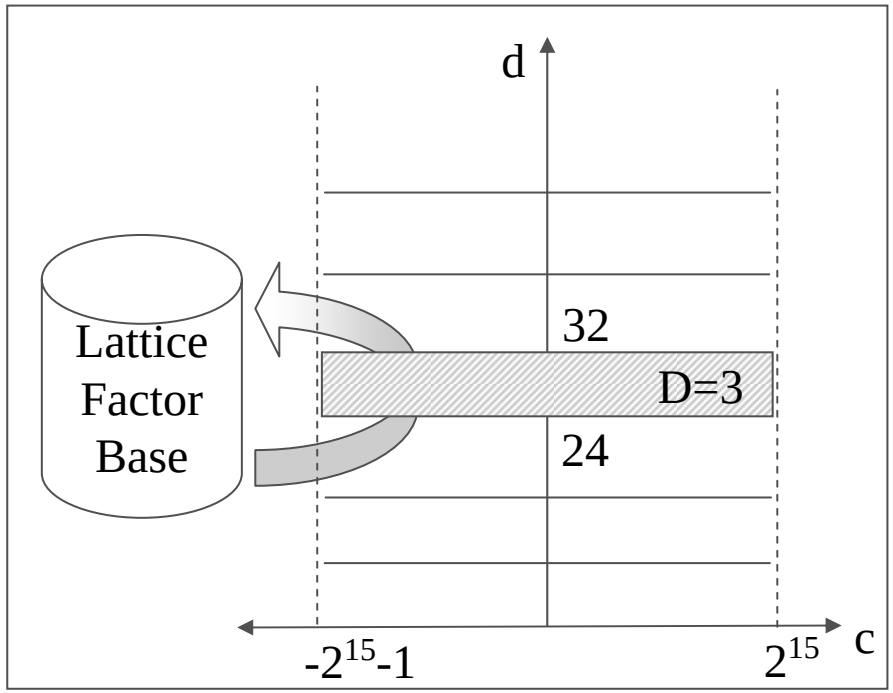
3. How to sequence the Lattice Points (2)

FUJITSU

Sieving Method

Memories 512KB in FPGA is assined 2^{19}-sieving points (-2^{15}～$2^{15}-1$, 8*D～8*D+7) (D=0,…,8191)

1. For each i=0,...,11, execute the step 2
2. Execute the step 3-6 for each lattice Base in the data base SBi_j ($j = D \mod 2^i$).
3. If d ≧ 8D then goto step 6.
4. Add log(p) to the sieving point according to the coordidate (c,d). [Sieving]
5. Calculate new coordinate (c', d') by using the Franke&Kleinjung Method , and go to the step 3.
6. Store the lattice base $[(c',d'),(c_1,d_1),(c_2,d_2)]$ to the tail of the data base Si_k, where $k = d/8 \mod 2^i$. [Sorting]



1. Caluculation of the Lattice Base

FUJITSU

Input Parameter

(q, r) : Special-Q (One of the Factor base。q:prime, r: integer smaller than q)
(a1, b1), (a2, b2) : Reduced base calculated by Special-Q
(p, t) : a Factor Base

Calculation

1. Translate (p,t) into the Lattice space generated by (a1,b1),(a2,b2)

$$s = (a_1 + b_1 t)(a_2 + b_2 t)^{-1} \bmod p$$

⇒ **Extended Binary GCD**

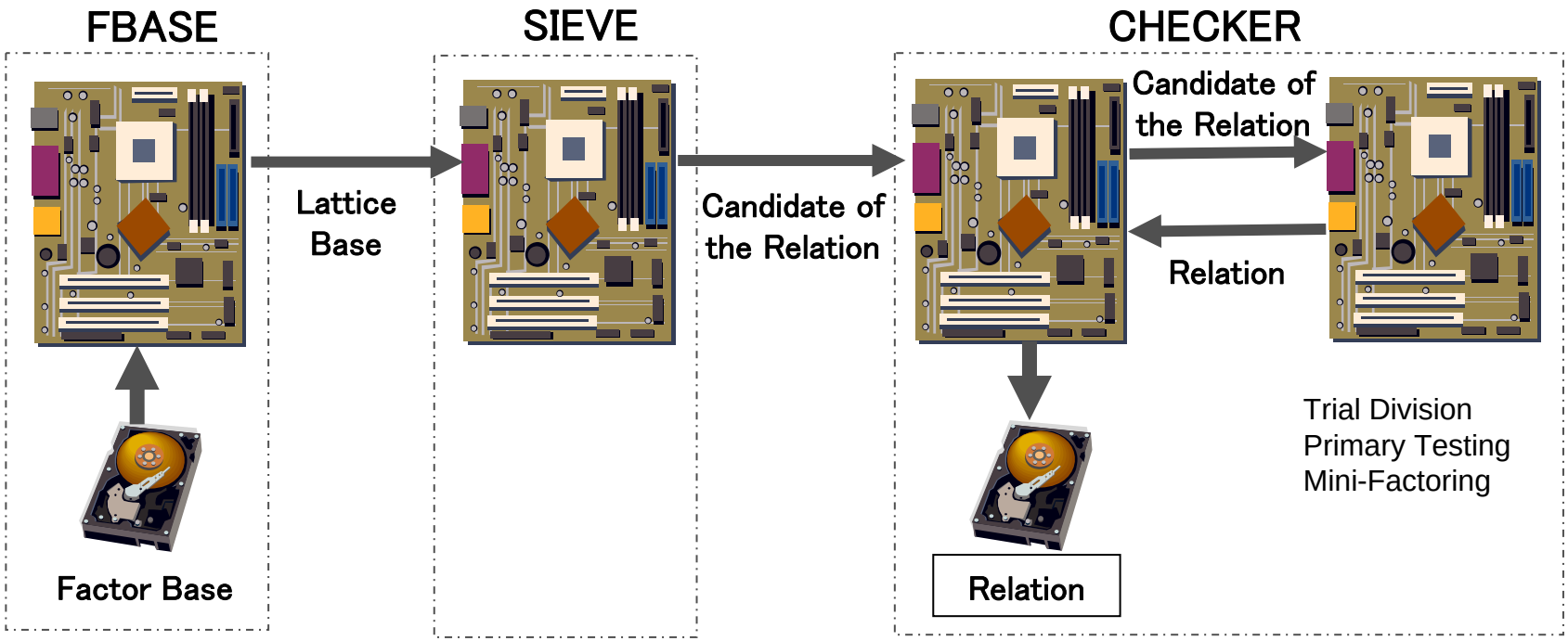
2. Calculate Lattice Base (α, β), (γ, δ) by using Franke & Kleijung method

```
(i[0], j[0]) = (-p, 0), (i[1], j[1]) = (s, 1),
Repeat the following from k=1
    (i[k+1] , j[k+1] ) = (i[k-1] , j[k-1] ) + a_k(i[k], j[k]),
    a_k =[|i[k-1] /i[k]|],
    If |i[k]| < I & |i[k-1] | ≧ I then quit from the loop.
    k = k+1,
Find the smallest "a"  such that |i[k-1] | - a|i[k]| < I
If k is even
    (α,  β) = (i[k-1] , j[k-1] ) + a(i[k] , j[k] )
    (γ,  δ) = (i[k] , j[k] )
If k is odd
    (α, β) = (i[k] , j[k])
    (γ, δ) = (i[k-1] , j[k-1] ) + a(I[k] , j[k] )
```

⇒ **Extended Lehmer GCD (a part)**

Sieving Hardware

- Combination of the three kind of the devices
  1. Generator of the Lattice Base (FBASE)
  2. Lattice Siever (SIEVE)
  3. Checker of the relation (CHECKER)
- Flow of the calculation



Results of the Implementation

FUJITSU

- Maximum input : 768-bit composit number
- Size in FPGA

	SLICE (%)	RAM(%)	LUT(%)	Register(%)
FBASE	99.998%	89.0%	80.1%	48.8%
SIEVE	40.9%	98.8%	32.1%	19.6%
CHECKER	78.0%	40.0%	45.0%	42.0%
Total	89,088	336	178,176	178,176

- Throughput
  - C128

Computation	Device	Throughput	Comment
Lattice Base	FPGA	0.0992 sec	For one Special-Q
Sending Lattice Base	DIO	0.1080 sec	For one Special-Q
Initial Setting	CPU	7.959 sec	Sieving Area=(2^{13}, 2^{13})
Sieving	FPGA	10.915 sec	Sieving Area=(2^{13}, 2^{13})
Sending Relation	EtherNet	0.237 sec	26528Byte

  - RSA768

Computation	Device	Throughput	Comment
Lattice Base	FPGA	2.75 sec	For one Special-Q
Sending Lattice Base	DIO	2.40 sec	For one Special-Q
Initial Setting	CPU	61.49 sec	Sieving Area=(2^{16}, 2^{16})
Sieving	FPGA	190.43 sec	Sieving Area=(2^{16}, 2^{16})
Sending Relation	EtherNet	0.044 sec	3812Byte

Security Evaluation of RSA

FUJITSU

- Estimation for factoring 768 bit RSA key
  - Timing for one relation＝3.920sec.
  - The number of requirement of the relations= 2.17×10^9

  ⇒ 3.920 sec. × (2.17×10^9) ＝ About 270 years
- Comparison with the previous HW in 2005
  - In the case of RSA768, this HW achived 38 times good performance compared with the previous one(※1).

	Line Sieve	Lattice Sieve	Comments
Relation ratio	1	23.43	By effect of Line Sieve→Lattice Sieve
Throughput (ms)	49.92	30.75	By Minor Improvement
Comparison	1	38.04	23.43 × (49.92/30.75)

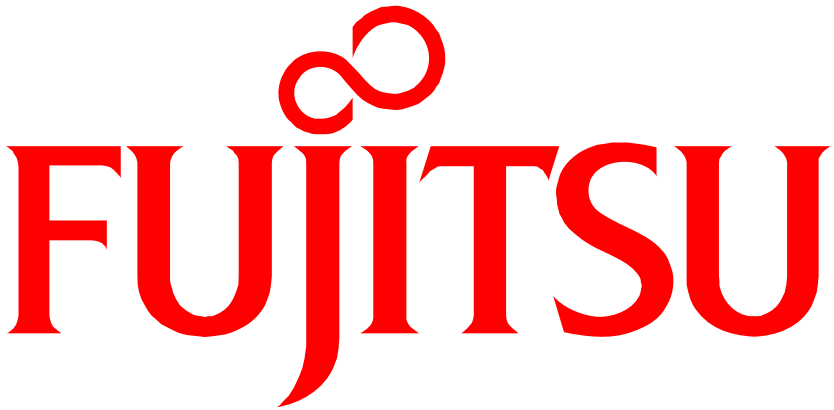
- Comparison with the Software
  - According to the Software evaluation(※2), it will take 1108 years for one PC (AMD Opteron). Then, the CAIRN3 is about 4.1 times faster than PC.

※1 See also CHES2007 (2007.9.13)
※2 From the Kleinjung Report (CRYPTREC Report 2006)

Conclusion

FUJITSU

- Implementation of CAIRN3
  - It succeeded in the development of the Relation Search HW CAIRN3 by using Lattice Sieving (The 1st in the World!)
  - In the case of RSA768, CAIRN3 achived 38 times good performance compared with CAIRN2 (World Record!)
- Security Evaluation of RSA
  - We estimated that it will take about 270 years for factoring the 768-bit RSA key by using ond CAIRN3.
  - For 1024-bit RSA, we concluded that the development of the sieving hardware is almost impossible in a current technologies we have, because it is necessary to solve the following several difficult problems.
    - Require at least 19.4GB memory and large-scale memory control circuit,
    - Require High-speed data transfer circuit between each devices,
    - Require Large-scale data sorting circuit, etc…
  - The 1024-bit RSA would not become insecure for several years, even if we have used HW which is based on a current technologies and practical resources.



THE POSSIBILITIES ARE INFINITE