



PROOFS, August 24, 2013

A hierarchical graph-based approach to generating formally-proofed Galois-field multipliers

Kotaro Okamoto, Naofumi Homma, and Takafumi Aoki
Tohoku University, Japan

Arithmetic algorithms over Galois fields

- Demands of high security and reliable systems
 - Cryptography, Error correction code
 - Arithmetic operations over **Galois Fields (GF)**



- Arithmetic algorithms
 - Hardware algorithms for arithmetic operation
 - Determine the performance of arithmetic circuits

There are two major difficulties in designing arithmetic algorithms based on Galois fields

Design issues

■ Lowest-level description using **logical expressions**

- Difficult to describe GF arithmetic algorithms by conventional HDLs

e.g., $GF(2^{16})$ multiplier

```
out0[0] = ((((((in0[0] & in1[0]) ^ (in0[15] & in1[1])) ^ ((in0[14] &
in1[2]) ^ (in0[13] & in1[3]))) ^ (((in0[12] & in1[4]) ^
(in0[11] & in1[5])) ^ ((in0[10] & in1[6]) ^ (in0[9] &
      :
in0[14]) ^ in0[12]) & in1[15]]))));
```

■ Verification using logic simulation

- Require a huge simulation time especially for **arithmetic circuits with large operand lengths**
 - Larger-scale multipliers than $GF(2^{32})$

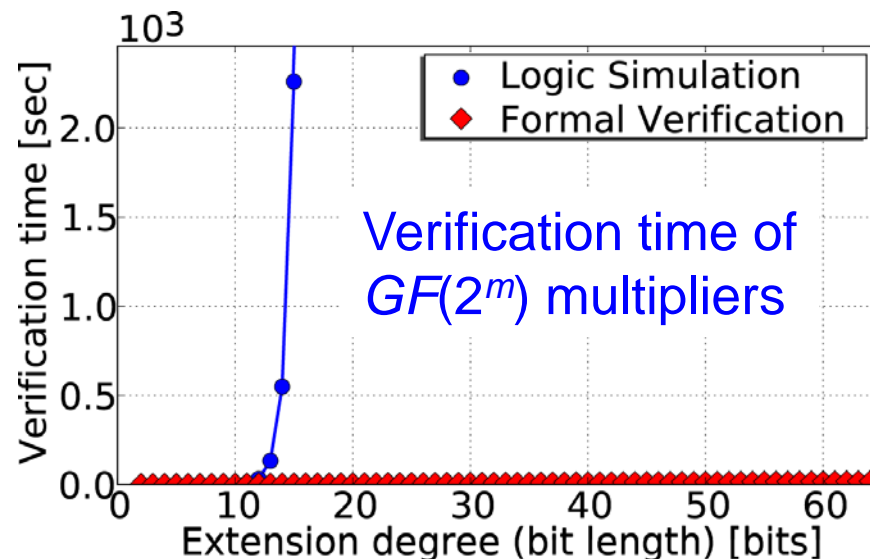
Graph-based approach

■ Galois-Field Arithmetic Circuit Graph: GF-ACG

- Represent a GF circuit using arithmetic equations based on GFs
- Hierarchical representation

■ Formal verification using computer algebra

- Gröbner basis
- polynomial reduction



This work

- Application to automatic generation system
 - Galois-Field Arithmetic Module Generator: GF-AMG
 - System producing **formally-proved** $GF(2^m)$ parallel multiplier for **any irreducible polynomial**
 - Mastrovito and Massey-Omura parallel multipliers



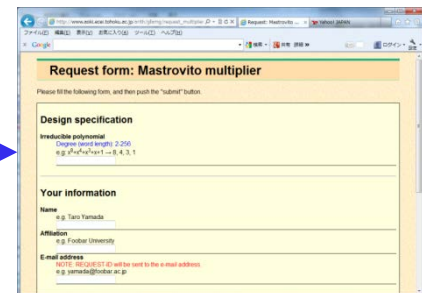
Designers

Design specification
Irreducible polynomial

```
module SD_MULTIPLIER(P, X, Y);
output TC P;
logic TC W, Y;
constraint begin
P.high = 16; P.low = 0;
X.high = 7; X.low = 0;
Y.high = 7; Y.low = 0;
end;
assertion P == X * Y;
structure begin
wire SD1[8];
wire SD2[8];
wire SD3[8];
constraint begin
S1.high = 3; S1.low = 0;
S2.high = 3; S2.low = 0;
for (i, 0, 3) begin
PP[i].high = 1;
end;
F.high = 15; F.low =
end;
SDOEN_ENCODE U0 (S,Y);
PPD ACCUMULATE U2 (P,PP);
SDTTC U3 (P,P);
end;
endmodule
```

Verified HDL codes
GSIS, TOHOKU UNIVERSITY

GF-AMG



Approach
based on
GF-ACGs



Outline

- Background
- Galois-Field Arithmetic Circuit Graph: GF-ACG
- Hierarchical design of Mastrovito multiplier
- Galois-Field Arithmetic Module Generator: GF-AMG
- Conclusion

Extension field

- Galois field of order p^m : $GF(p^m)$ p : prime number
- Each field element is a polynomial over $GF(p)$
- Addition and multiplication are performed modulo **irreducible polynomial IP** of degree m

e.g., $GF(2^2) = \{0, 1, \beta, \beta+1\}$, $IP = \beta^2 + \beta + 1$

Addition over $GF(2^2)$

+	0	1	β	$\beta+1$
0	0	1	β	$\beta+1$
1	1	0	$\beta+1$	β
β	β	$\beta+1$	0	1
$\beta+1$	$\beta+1$	β	1	0

Multiplication over $GF(2^2)$

\times	0	1	β	$\beta+1$
0	0	0	0	0
1	0	1	β	$\beta+1$
β	0	β	$\beta+1$	1
$\beta+1$	0	$\beta+1$	1	β

GF-ACG: Galois-Field Arithmetic Circuit Graph

$$\text{GF-ACG: } G = (N, E)$$

■ N : set of nodes

□ Node: $n = (F, G')$

– F : function (GF equation)

– G' : internal structure
(GF-ACG)

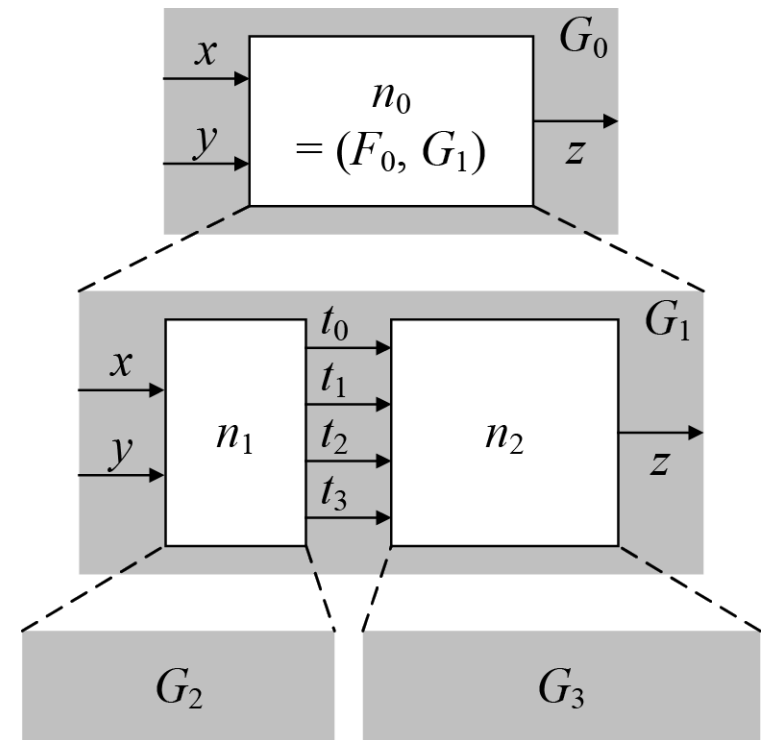
■ E : set of directed edges

□ Directed edge: $e = (n_s, n_d, x)$

– n_s : source node

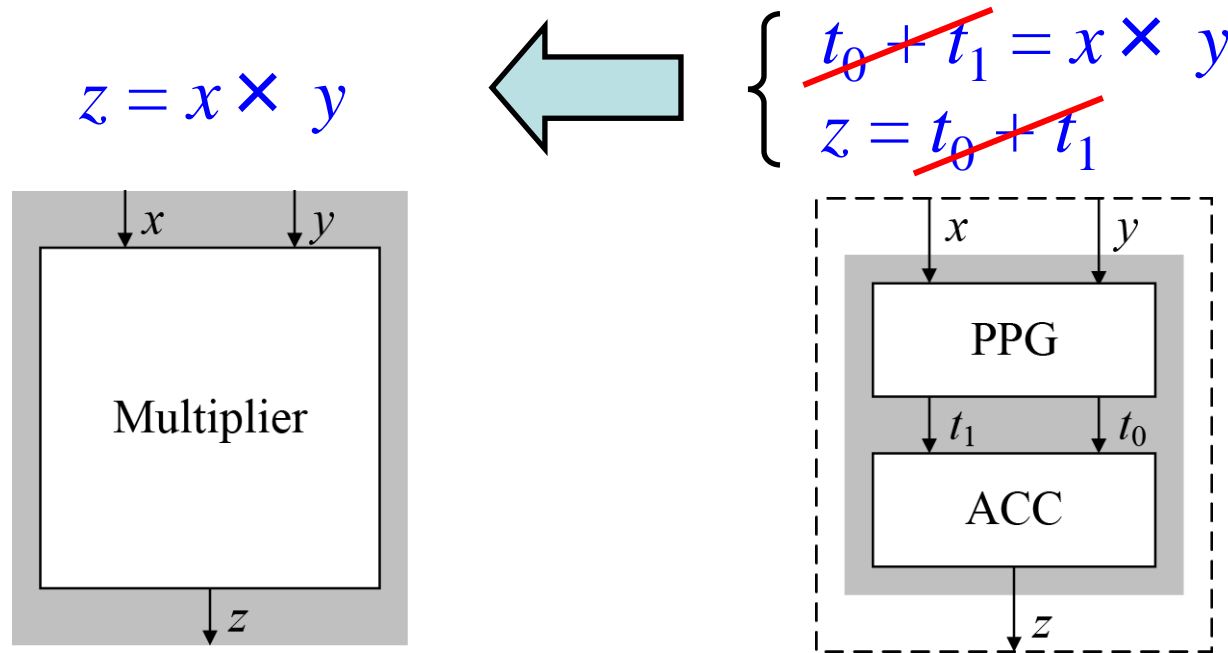
– n_d : destination node

– x : GF variable



Formal verification of GF-ACGs

- Verification is done by checking equivalence between the function and the internal structure
 - Function is correct if same function is derived from internal structure



Solve simultaneous equation by computer algebra



Outline

- Background
- Galois-Field Arithmetic Circuit Graph: GF-ACG
- Hierarchical design of Mastrovito multiplier
 - Typical $GF(2^m)$ parallel multiplier
- Galois-Field Arithmetic Module Generator: GF-AMG
- Conclusion

Mastrovito multiplier

■ Feature

- $GF(2^m)$ parallel multiplier
- **Smallest area**

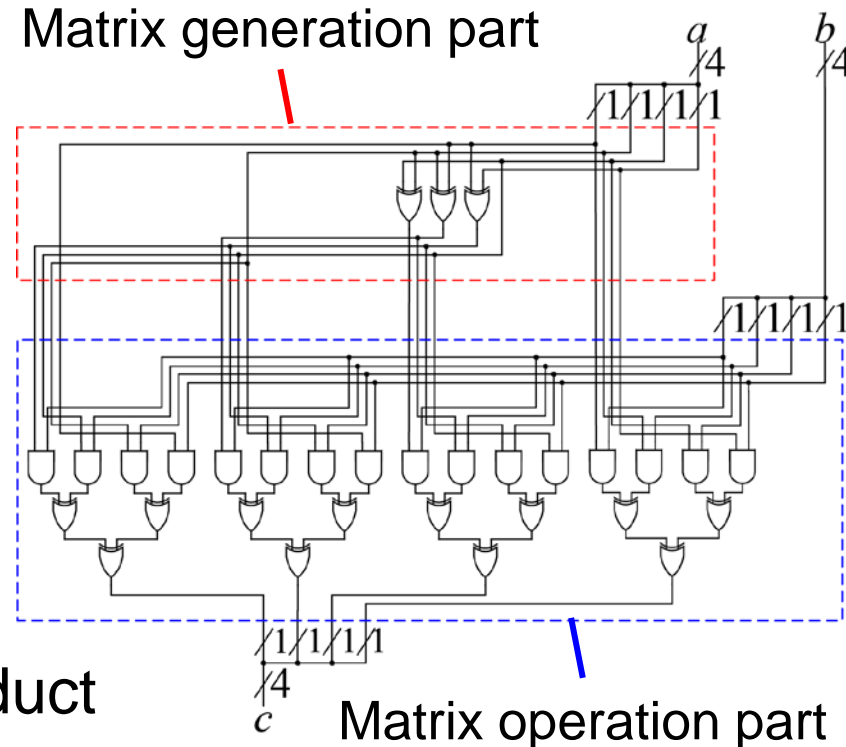
■ Structure

- Matrix generation part
 - Generation of matrix Z from the input a
- Matrix operation part
 - Calculation of inner product of Z and the other input b

e.g., $GF(2^4)$ multiplier

for $IP = \beta^4 + \beta + 1$

Matrix generation part



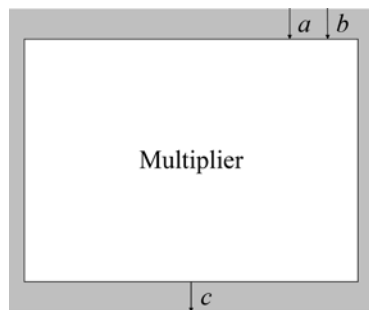
Hierarchical description for GF-ACG design

Why hierarchical description ?

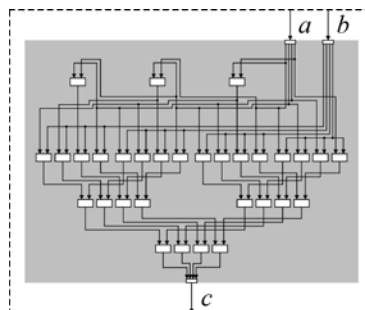
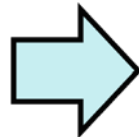
- Necessary to derive **hierarchical description** from original flattened description

e.g., $GF(2^4)$ multiplier

Top level description Flattened description



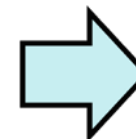
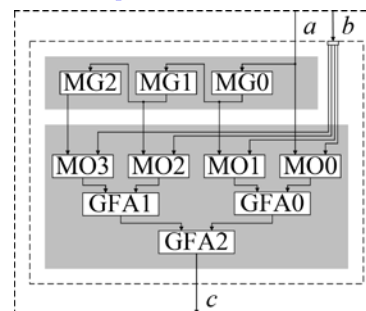
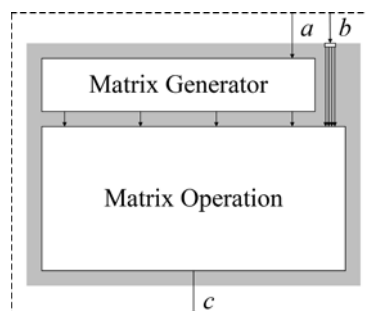
NG!



Number of variables increases exponentially against bit length

Hierarchical description

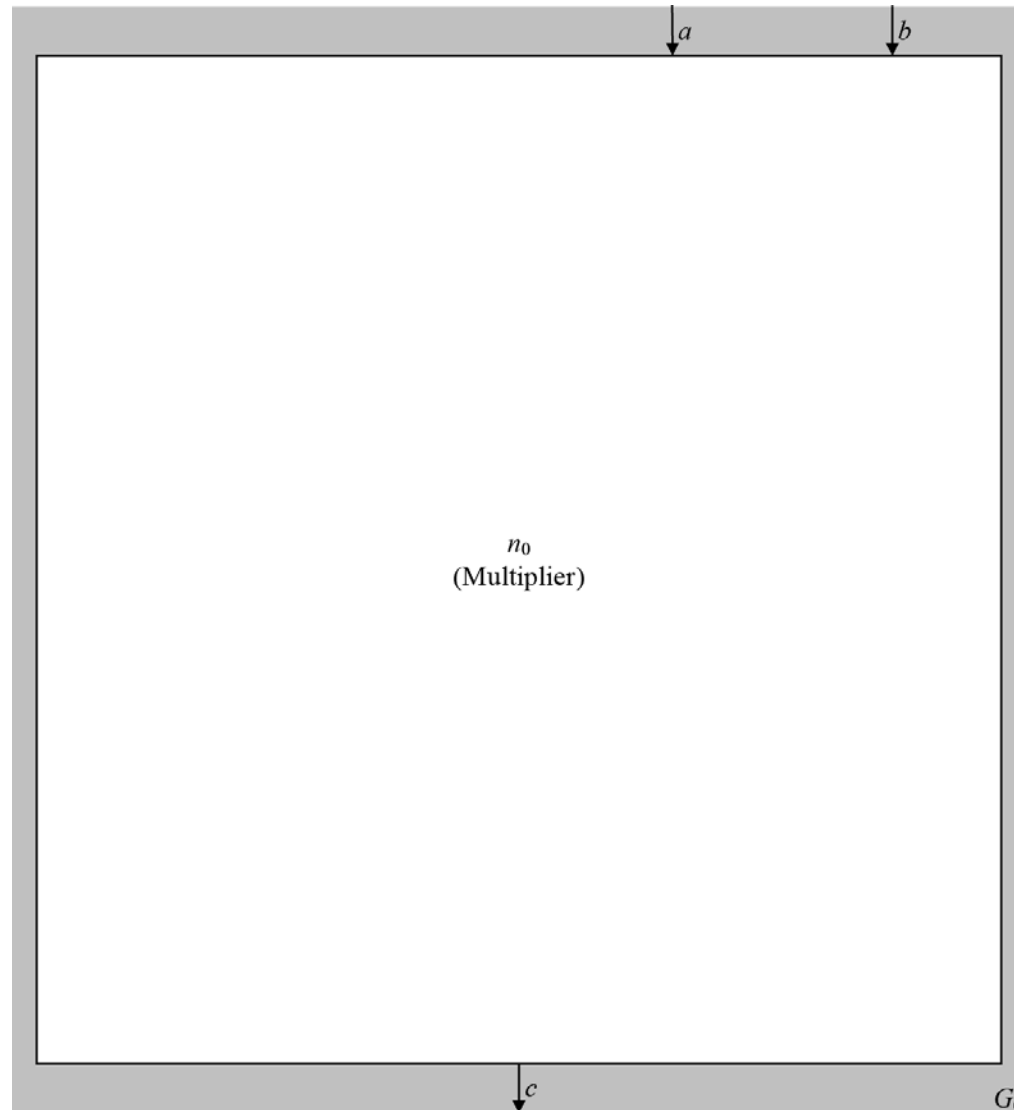
OK!



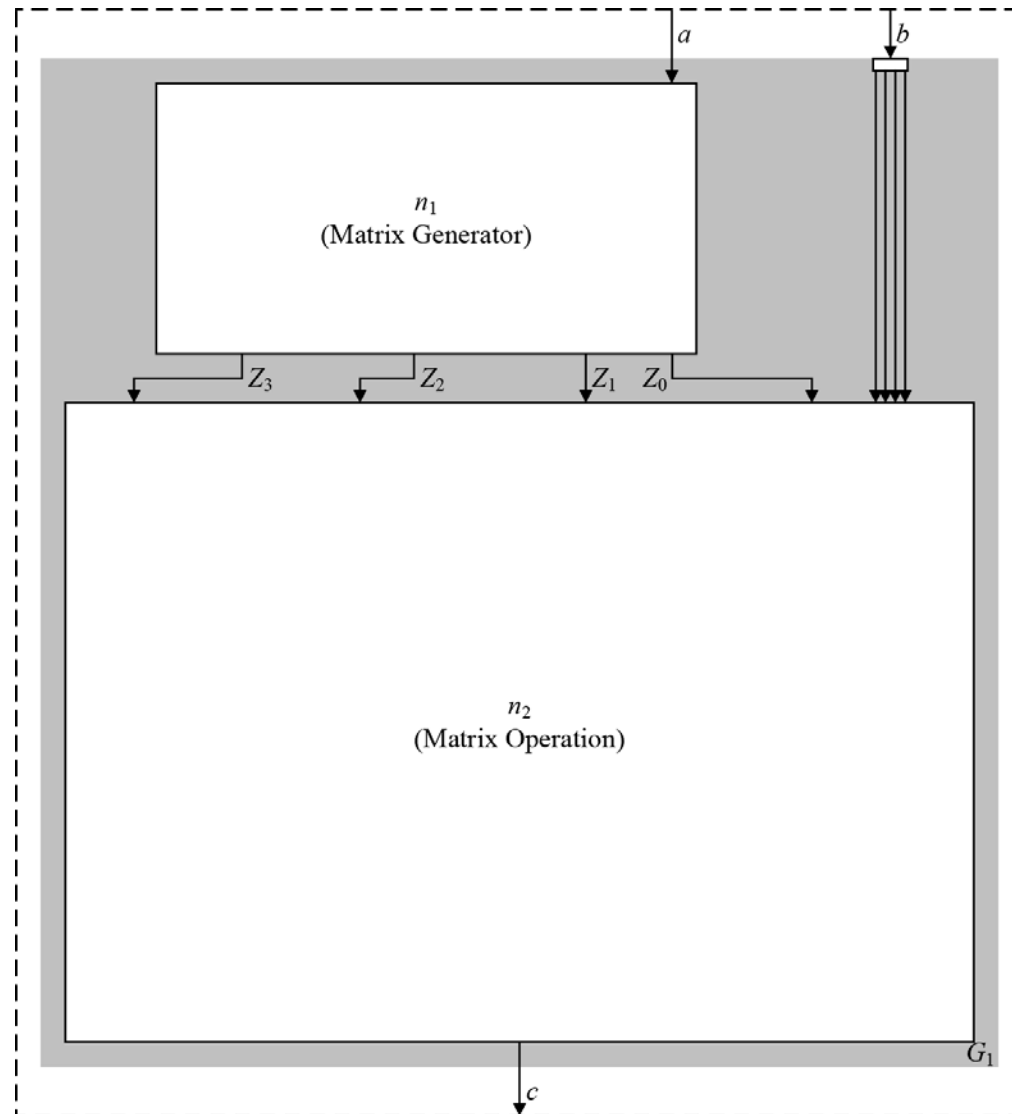
Nodes and functions for GF-ACG design

Node	Function
Multiplier	$c = a \times b$
└ Matrix Generator	$Z_i = a \cdot \beta^i, 0 \leq i \leq m-1$
└┬ MG	$Z_i = Z_{i-1} \cdot \beta$
└ Matrix Operation	$c = \sum_{i=0}^{m-1} Z_i \times (b_i^{(e)} \cdot \beta^{-i})$
└┬ MO	$w_i = Z_i \times (b_i^{(e)} \cdot \beta^{-i})$
└┬ GFA	$w_{m+i} = w_{2i} + w_{2i+1}$

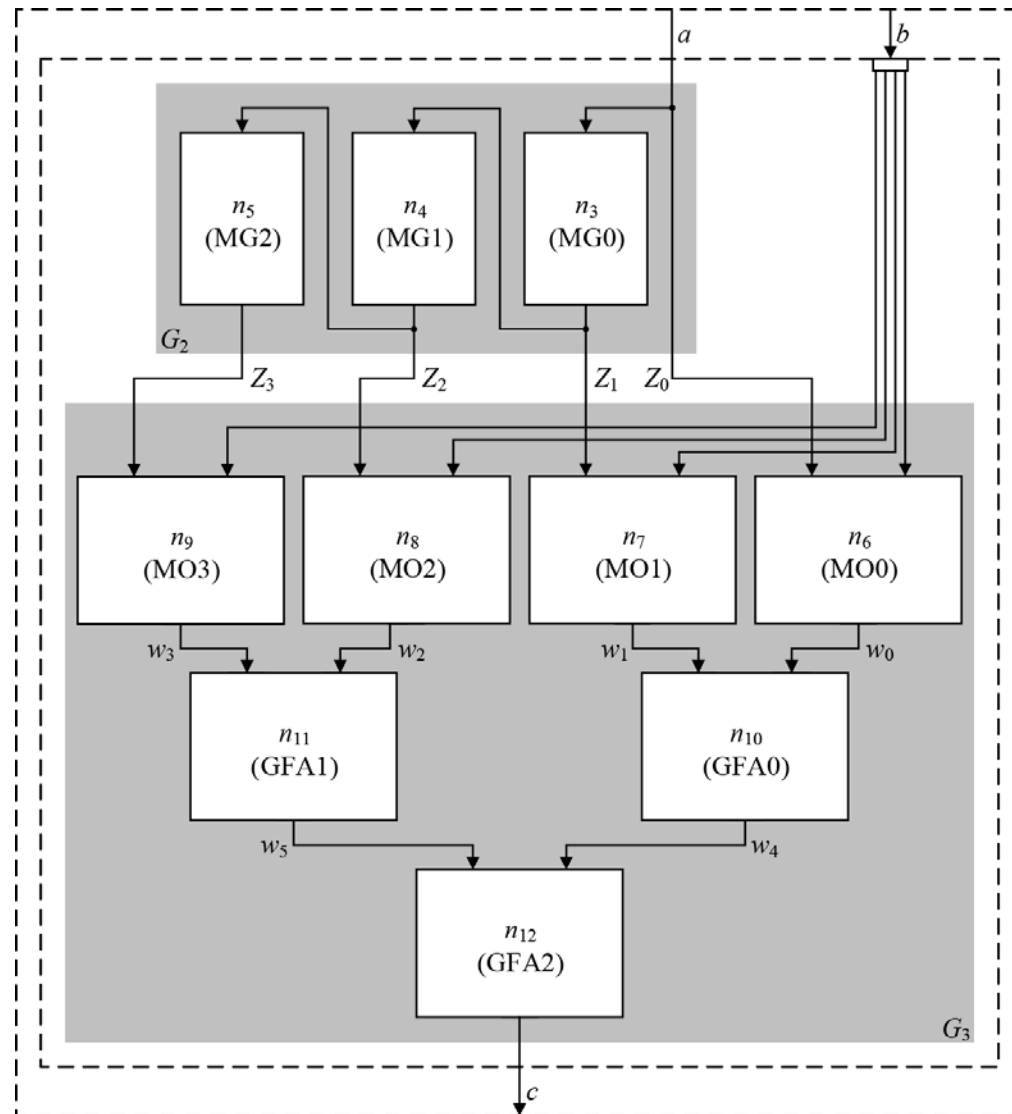
GF-ACG for $GF(2^4)$ Mastrovito multiplier



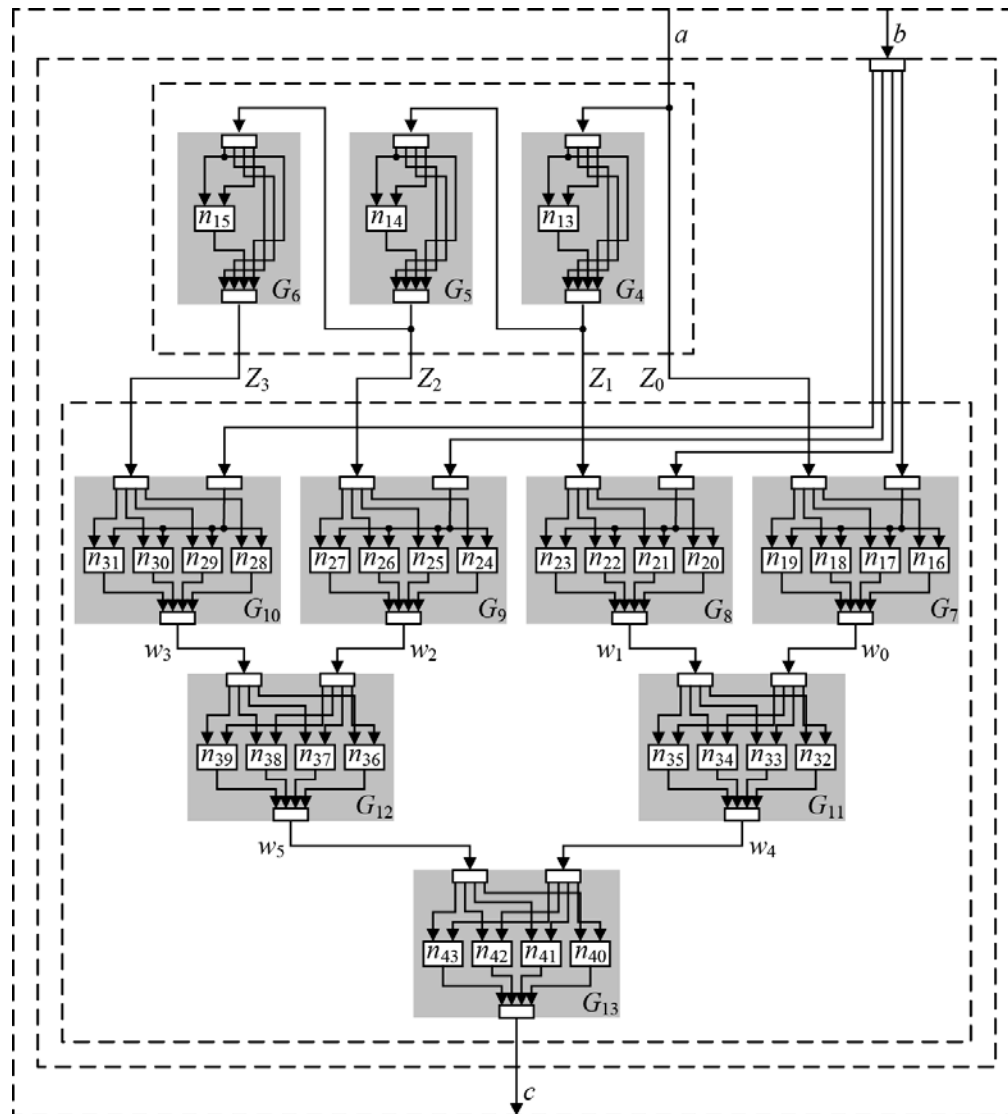
GF-ACG for $GF(2^4)$ Mastrovito multiplier



GF-ACG for $GF(2^4)$ Mastrovito multiplier



GF-ACG for $GF(2^4)$ Mastrovito multiplier





Outline

- Background
- Galois-Field Arithmetic Circuit Graph: GF-ACG
- Hierarchical design of Mastrovito multiplier
- Galois-Field Arithmetic Module Generator: GF-AMG
 - Application of GF-ACG approach
- Conclusion

$GF(2^m)$ multiplier generator on Website

■ Feature

- Automatic generation system of $GF(2^m)$ multipliers for **any irreducible polynomial IP**
- Generate only **formally-proved HDL codes**

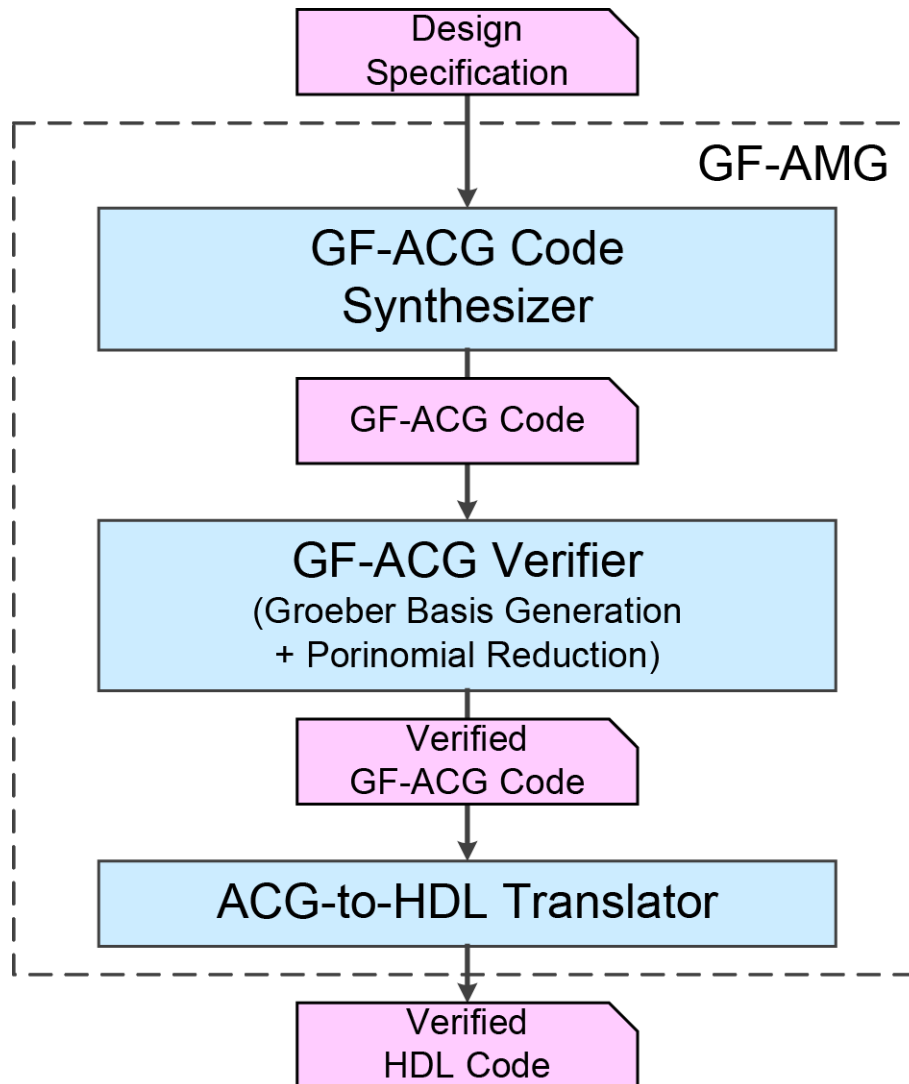
■ System specification

Multiplication algorithm	Degree for IP
Mastrovito algorithm	From 2 to 256
Massey-Omura algorithm	From 2 to 64

■ Available from website

<http://www.aoki.ecei.tohoku.ac.jp/arith/gfamg>

Block diagram of GF-AMG



Design Specification

Irreducible polynomial

GF-ACG Code Synthesizer

Generation of GF-ACG code according to design specification

GF-ACG Verifier

Formal verification of generated GF-ACG code

ACG-to-HDL Translator

Translation of GF-ACG code into equivalent HDL code

Verified Multiplier

Verilog-HDL code

Performance evaluation

Generation time of Mastrovito multiplier [sec]

	$GF(2^8)$	$GF(2^{16})$	$GF(2^{32})$	$GF(2^{64})$	$GF(2^{128})$
Logic simulation	0.279	9,330	N/A	N/A	N/A
Formal verification	3.374	5.188	9.487	19.55	52.61

Generation time of Massey-Omura parallel multiplier [sec]

	$GF(2^8)$	$GF(2^{16})$	$GF(2^{32})$	$GF(2^{64})$	$GF(2^{128})$
Logic simulation	0.460	N/A	N/A	N/A	N/A
Formal verification	3.618	5.482	16.24	372.5	34,263

Complete simulation of $GF(2^{32})$ multiplier was impossible

Complete verification of $GF(2^{128})$ multiplier was possible

Linux CPU: Intel Core2 Due E4600 2.40GHz, 7GB Memory
Formula manipulation software: Risa/Asir

Demonstration

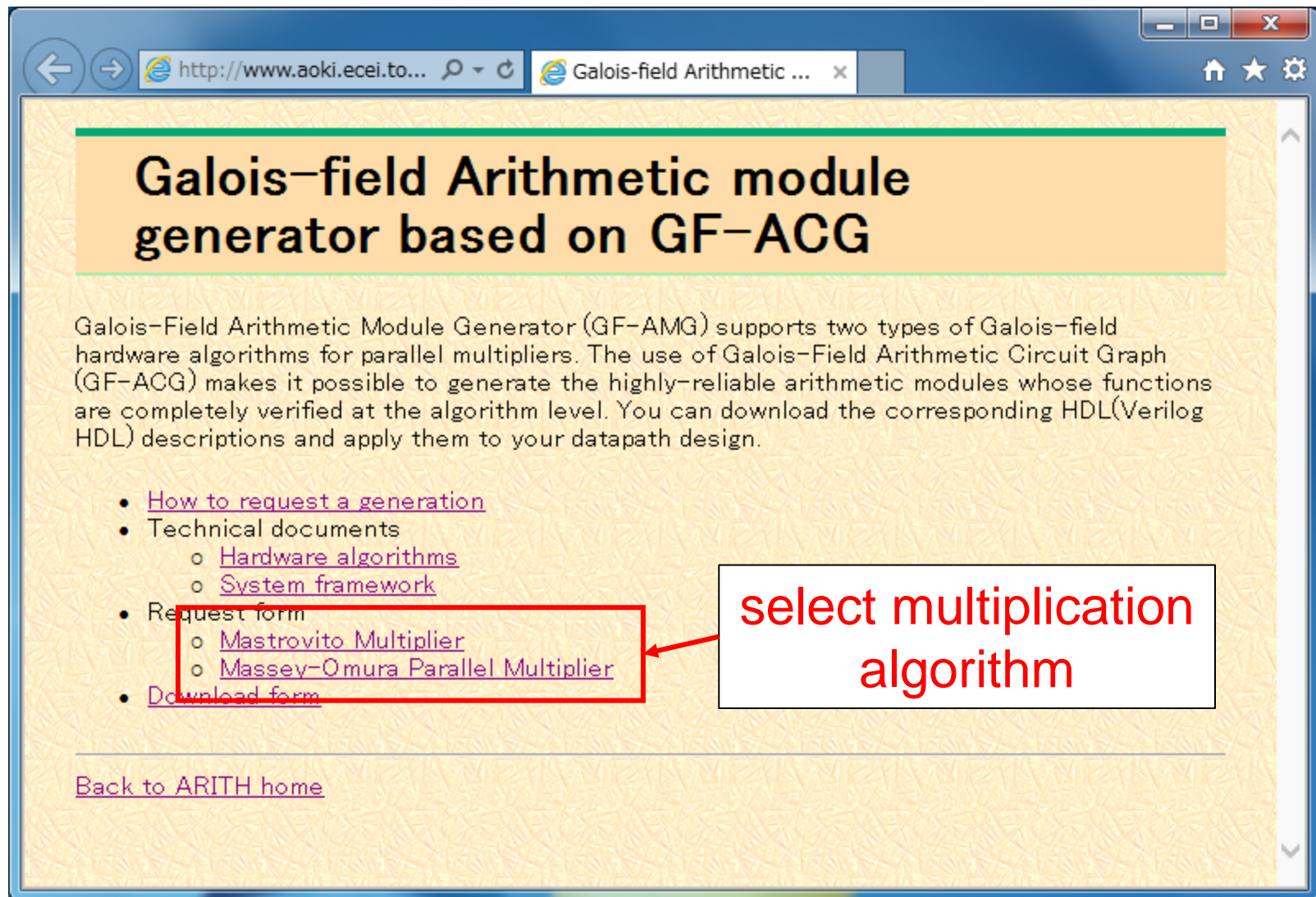
- ~~Activation of GF-AMC~~
- Stop of service for maintenance
 - Japanese holiday
- Available from August 26
- Explanation using some slides
 - Substitution for demonstration

Access web-page



<http://www.aoki.ecei.tohoku.ac.jp/arith/gfamg>

Website for GF-AMG



Galois-field Arithmetic module generator based on GF-ACG

Galois-Field Arithmetic Module Generator (GF-AMG) supports two types of Galois-field hardware algorithms for parallel multipliers. The use of Galois-Field Arithmetic Circuit Graph (GF-ACG) makes it possible to generate the highly-reliable arithmetic modules whose functions are completely verified at the algorithm level. You can download the corresponding HDL(Verilog HDL) descriptions and apply them to your datapath design.

- [How to request a generation](#)
- Technical documents
 - [Hardware algorithms](#)
 - [System framework](#)
- Request form
 - [Mastrovito Multiplier](#)
 - [Massey-Omura Parallel Multiplier](#)
- [Download form](#)

[Back to ARITH home](#)

select multiplication algorithm

Submission of generation request

Request form: Mastrovito multiplier

Please fill the following form, and then push the "submit" button.

Design specification

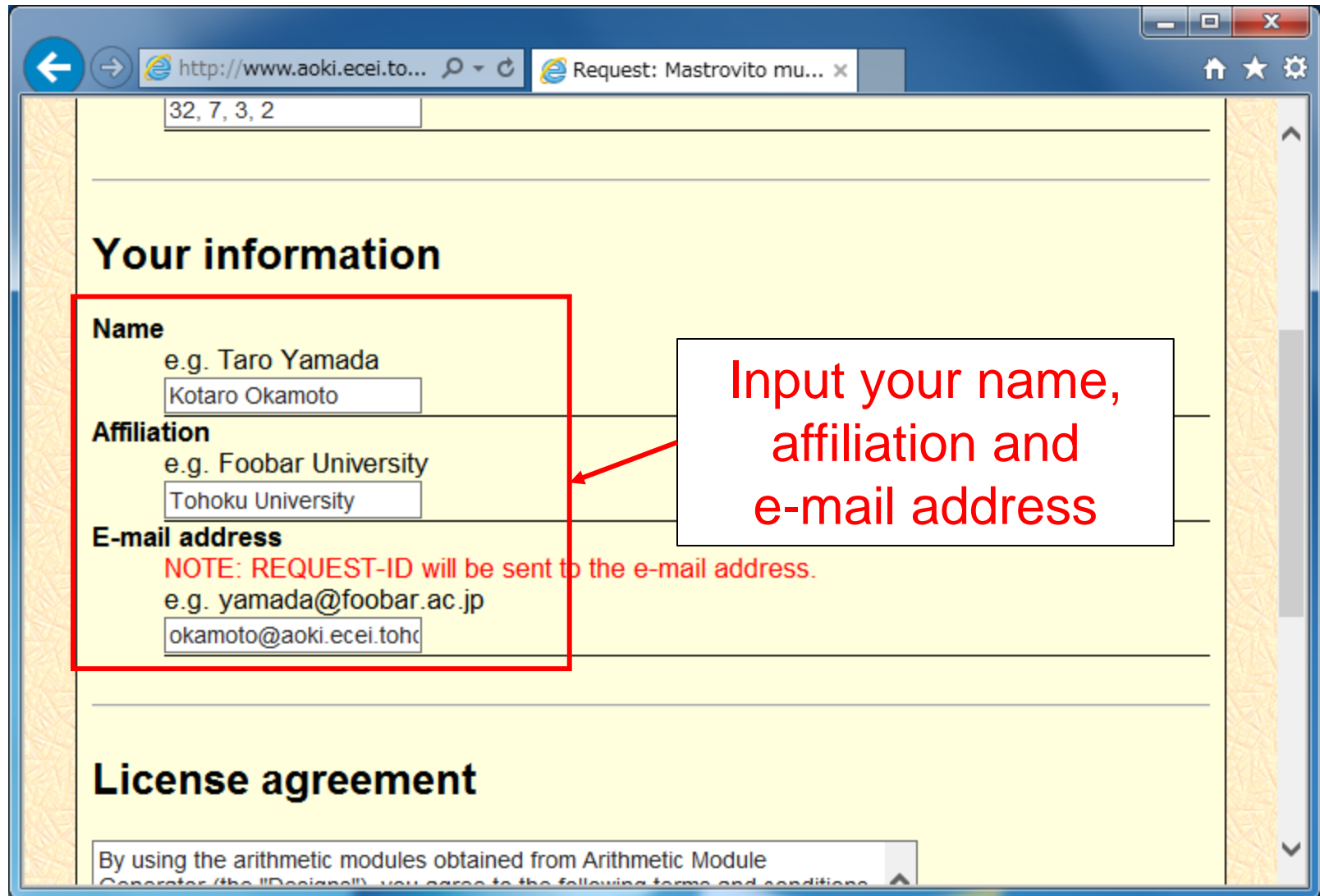
Irreducible polynomial
Degree (word length): 2-256
e.g: $x^8+x^4+x^3+x+1 \rightarrow 8, 4, 3, 1$

Your information

Name
e.g. Taro Yamada

Input irreducible polynomial

Submission of generation request



32, 7, 3, 2

Your information

Name
e.g. Taro Yamada

Affiliation
e.g. Foobar University

E-mail address
NOTE: REQUEST-ID will be sent to the e-mail address.
e.g. yamada@foobar.ac.jp

License agreement

Input your name, affiliation and e-mail address

Submission of generation request

The screenshot shows a web browser window with the address bar displaying `http://www.aoki.ecei.to...` and a tab titled "Request: Mastrovito mu...". The main content area is titled "License agreement" and contains the following text:

By using the arithmetic modules obtained from Arithmetic Module Generator (the "Designs"), you agree to the following terms and conditions.

The Designs are copyrighted information of Aoki Laboratory ("us"). Use of the Designs, with or without modification, is permitted provided that the following conditions are met:

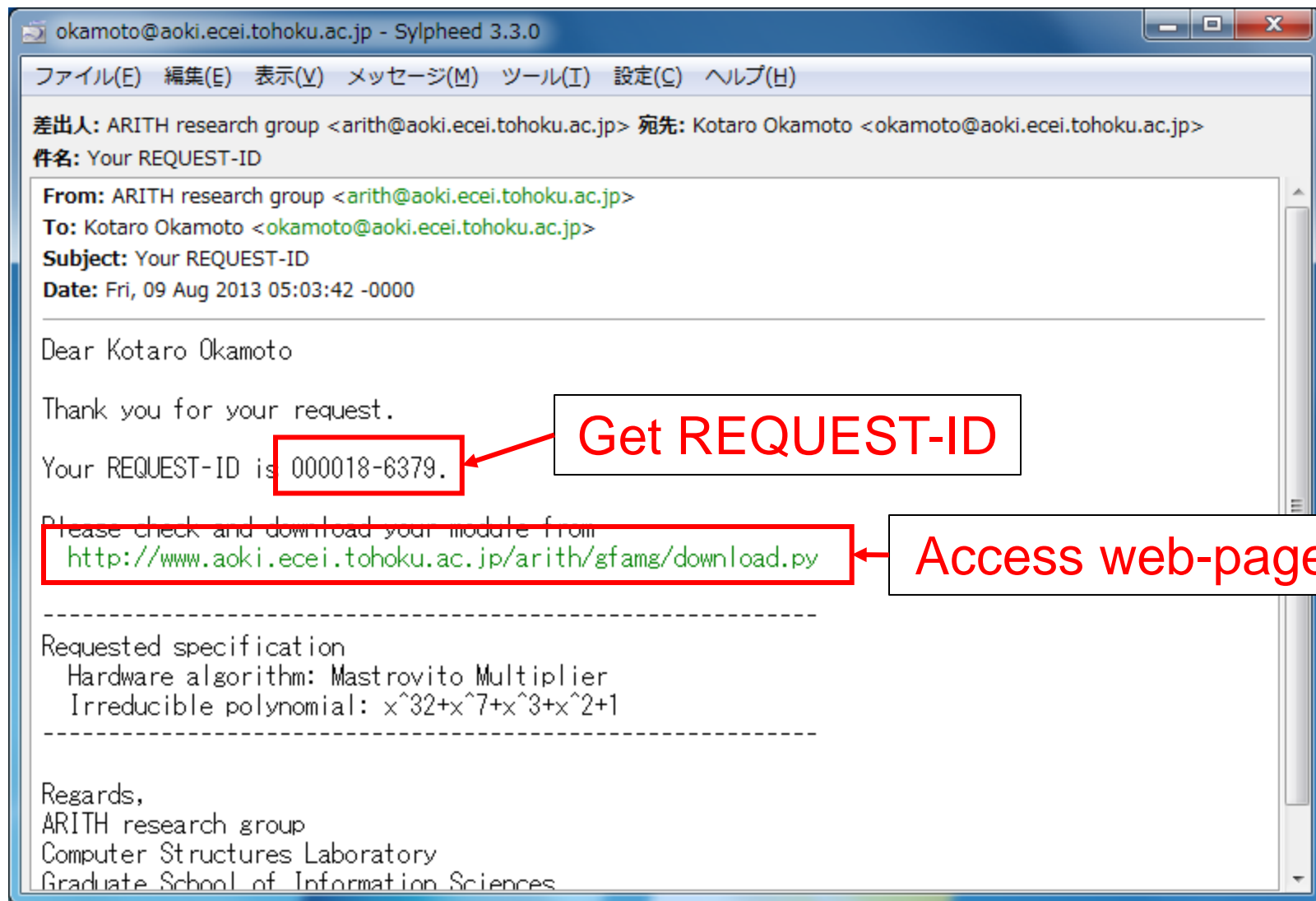
WE SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION DIRECT, INDIRECT, INCIDENTAL, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF THE

Below the text, there is a form with the question "Do you agree to the above terms and conditions?" and two radio buttons: "Yes" (selected) and "No". A red box highlights the "Yes" radio button, and a red arrow points from a white box labeled "Agree to license" to it.

Below the form is a "submit" button. A red box highlights the button, and a red arrow points from a white box labeled "Push 'submit' button" to it.

At the bottom of the page, there is a link: [Back to Galois-field Arithmetic Module Generator home](#)

Reception of email



Submission of REQUEST-ID

Download form

Please input the REQUEST-ID obtained in the [request page](#).

REQUEST-ID
e.g. 012345-6789
000018-6379 x

Input REQUEST-ID

submit

Push "submit" button

[Back to Galois-field Arithmetic Module Generator home](#)

Download

Following archive files contain the design you requested.
In Tar+Gzip archive: [download](#)
In Zip archive: [download](#)

Download

REQUEST-ID: 000018-6379

Status: Generation succeeded
Requested date: 2013-08-09 14:03:42 JST
Generated & verified date: 2013-03-07 07:48:52 JST

Requested specification

Hardware algorithm: Mastrovito multiplier
Irreducible polynomial: $x^{32}+x^7+x^3+x^2+1$

Conclusion and future work

■ Conclusion

- Hierarchical design of Mastrovito multiplier
- Application to automatic generation system
 - System specification

Multiplication algorithm	Degree for IP
Mastrovito algorithm	From 2 to 256
Massey-Omura algorithm	From 2 to 64

- Website for system

<http://www.aoki.ecei.tohoku.ac.jp/arith/gfamg>

■ Future work

- Development of advanced module generators for cryptographic datapaths with GF arithmetic circuits



END

Thank you for your attention