Parallel and Reconfigurable VLSI Computing (12)

Complexity of Logic Functions, and its Decomposition: Synthesis for an FPGA

Hiroki Nakahara

Tokyo Institute of Technology

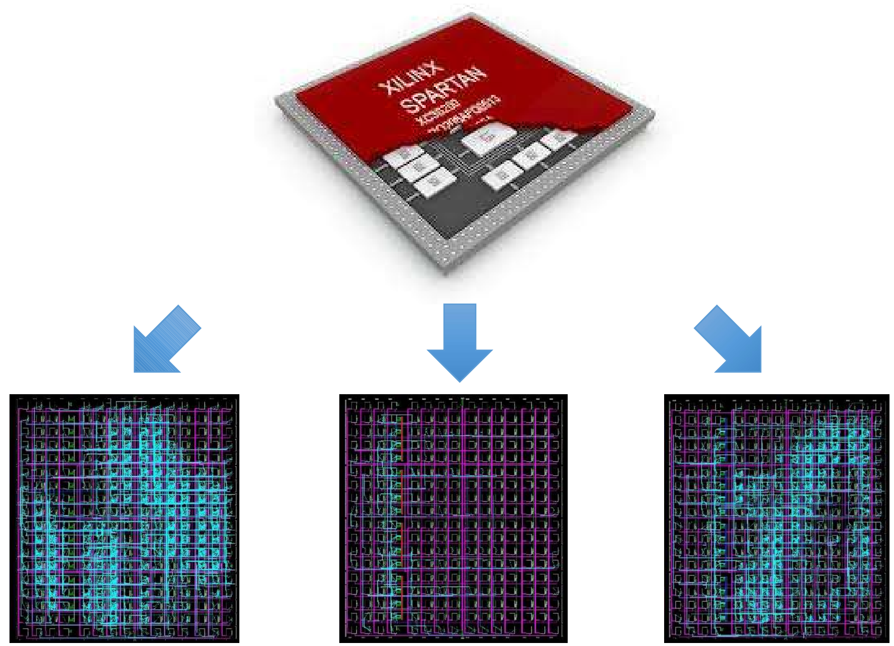
Outline

- Memory-based logic design for an FPGA
  - FPGA architecture
- Functional decomposition
  - Complexity of a logic function
- Case studies
  - Index generation function
  - Sum of weighted function
  - Residue Number System (RNS)
- Summary

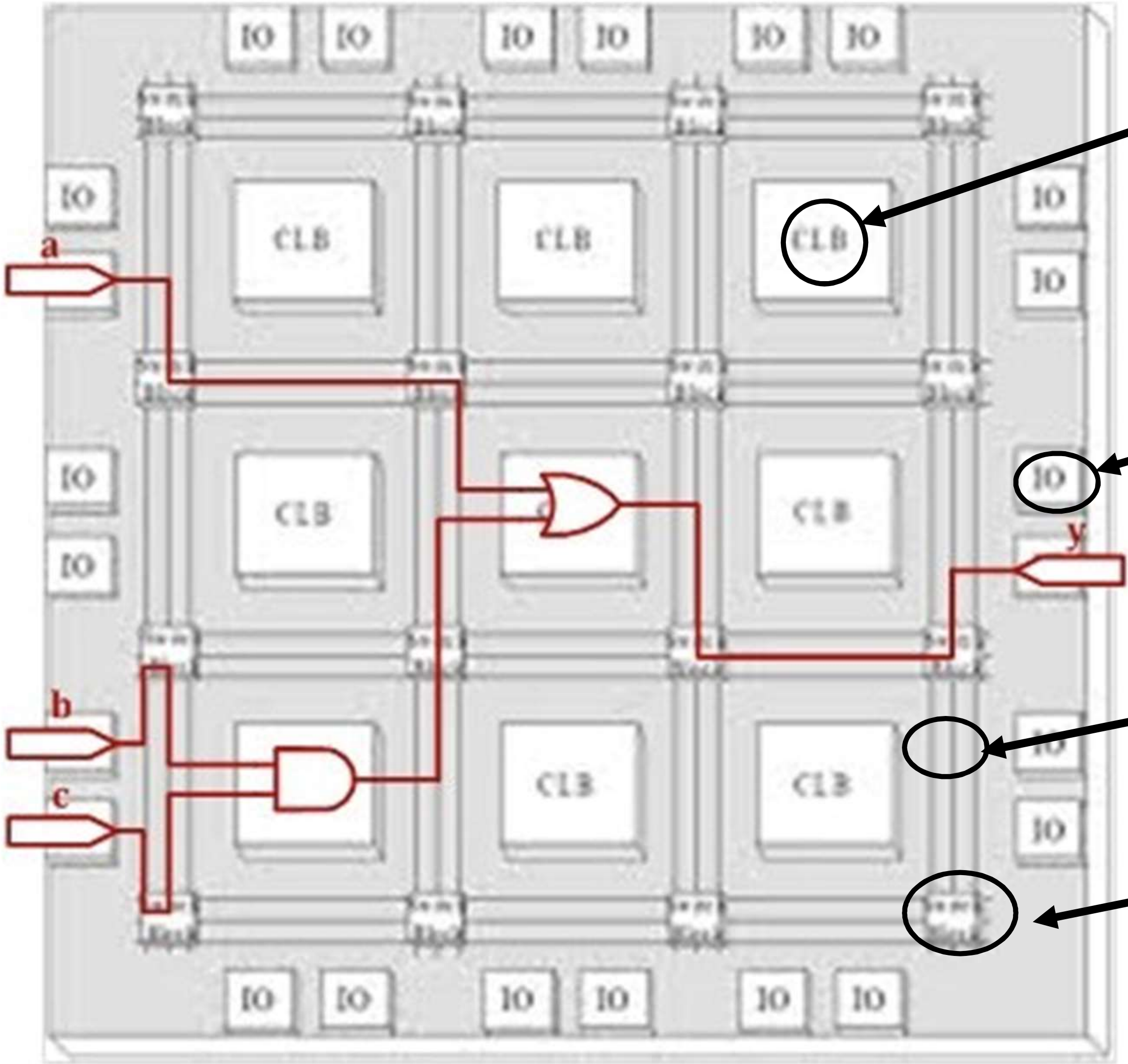
1 Memory-based logic design for an FPGA

FPGA

- Reconfigurable LSI or Programmable Hardware
- Programmable Logic Array and Programmable Interconnection
- Programmed by Reconfigurable Data

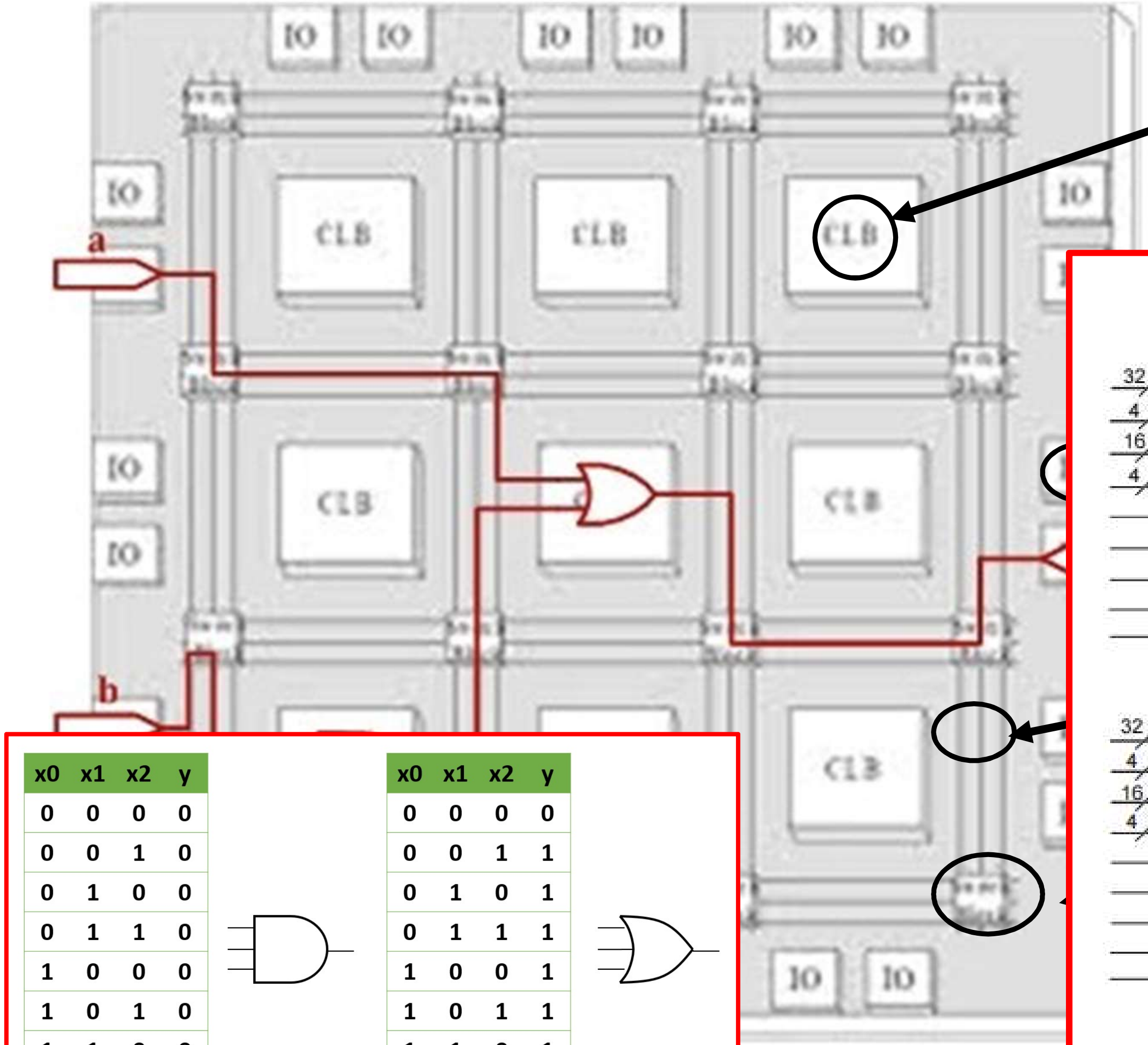


Island Style FPGA



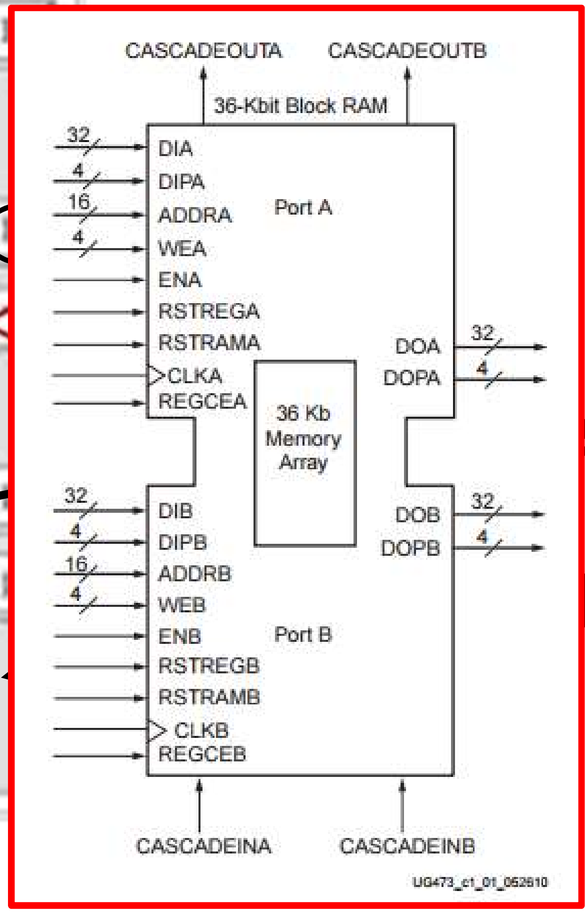
FPGA is a memory-based architecture

Configurable Logic Block (CLB)



x0	x1	x2	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

x0	x1	x2	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

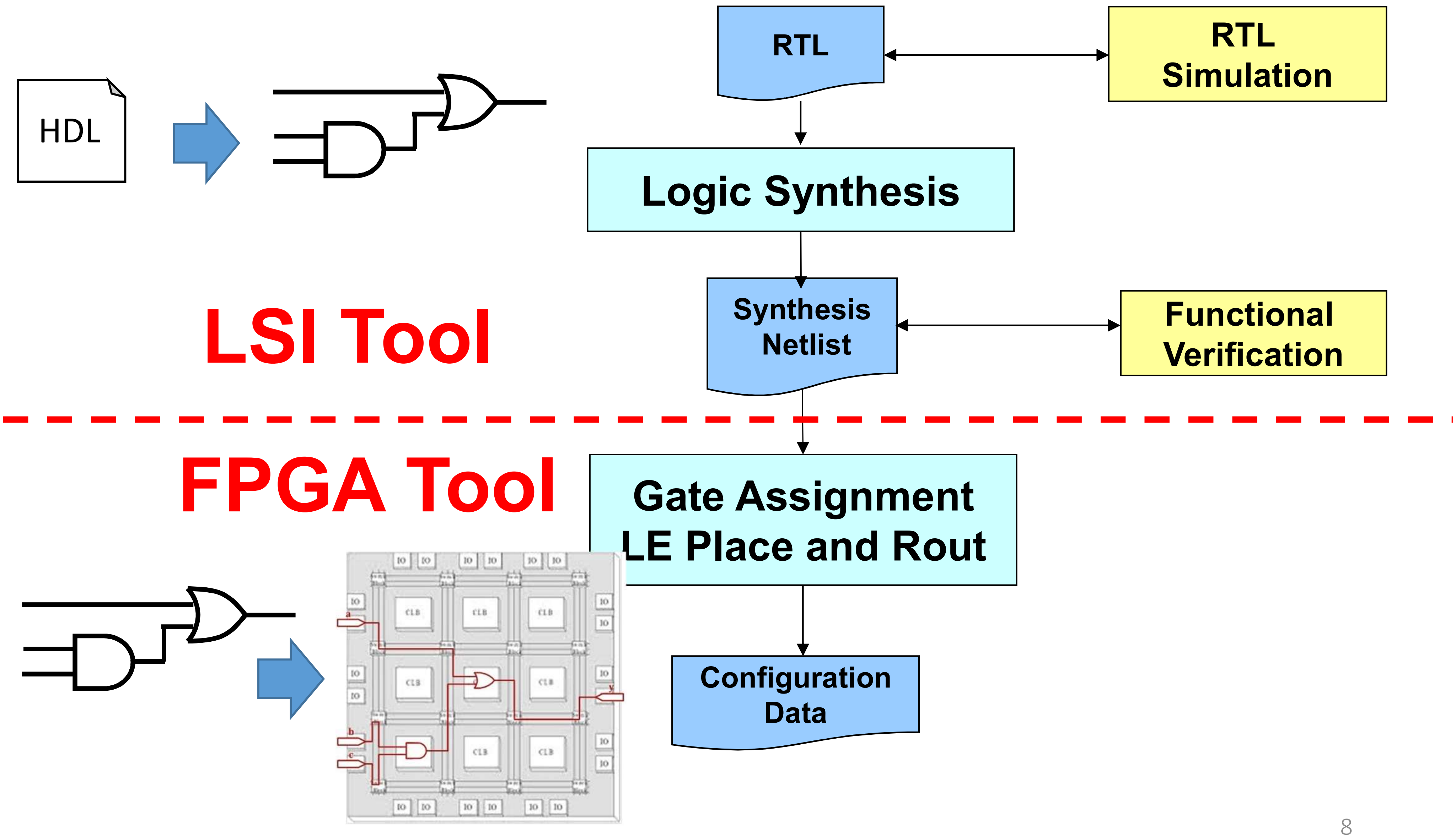


els

h Blocks

2. Functional decomposition for a Logic Function

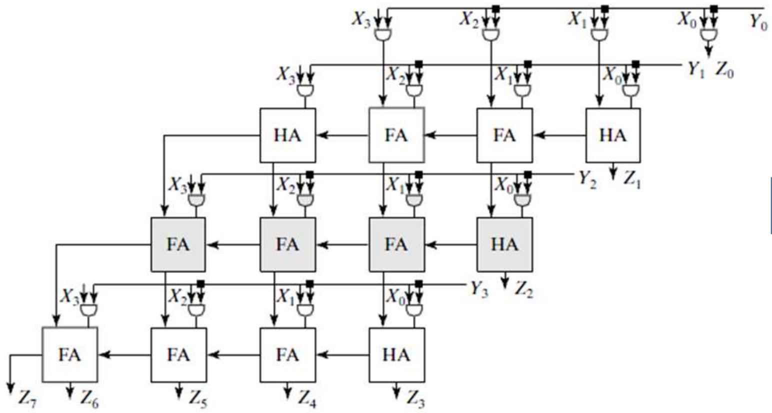
Standard FPGA Design

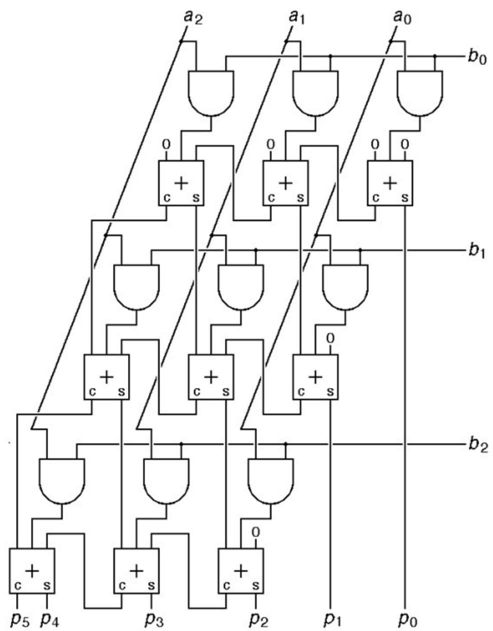


Logic Synthesis

- Synthesize from a given HDL specification to a Boolean network

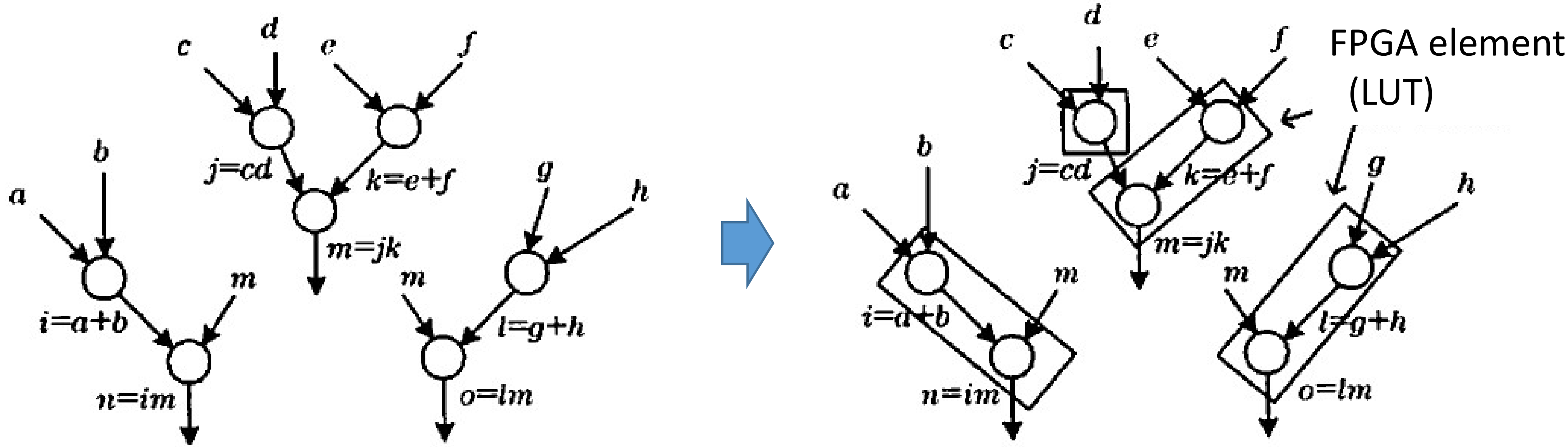
input [3:0]X,Y;
output [7:0]Z;
Z=X*Y





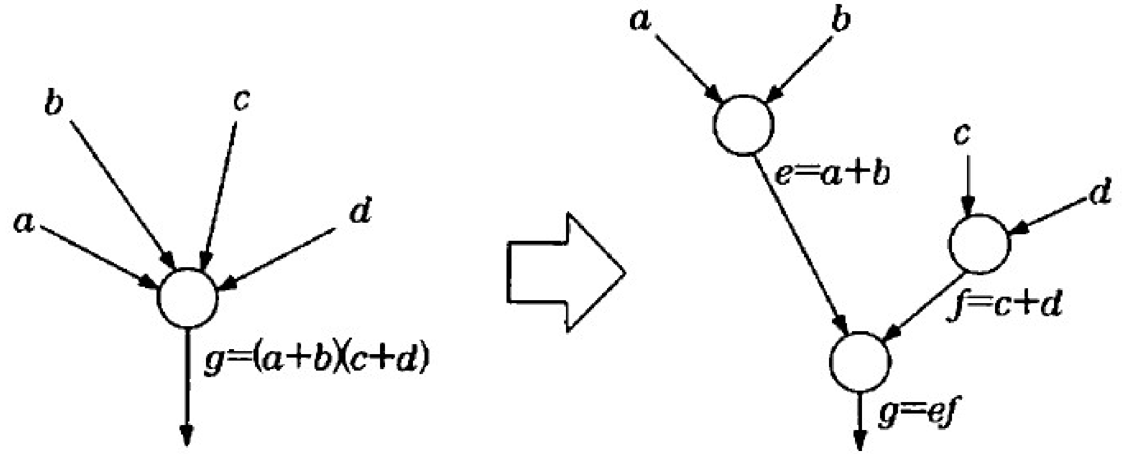
Technology Mapping

- A kind of a graph covering problem
- Goal: A depth optimized one by using a dynamic programming

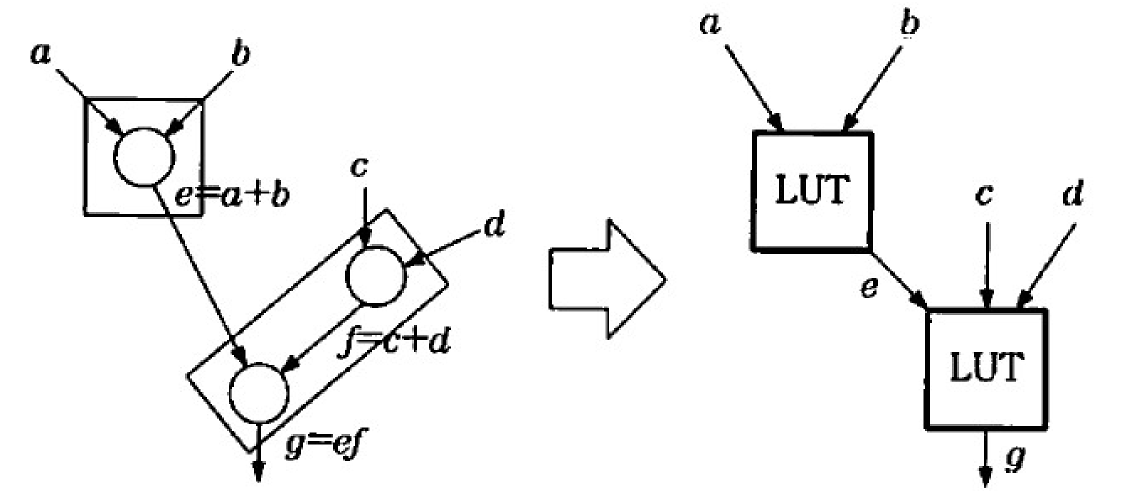


Decomposition and Covering

Decomposition



Covering



Functional Decomposition

Decomposition Chart

Bound Variables

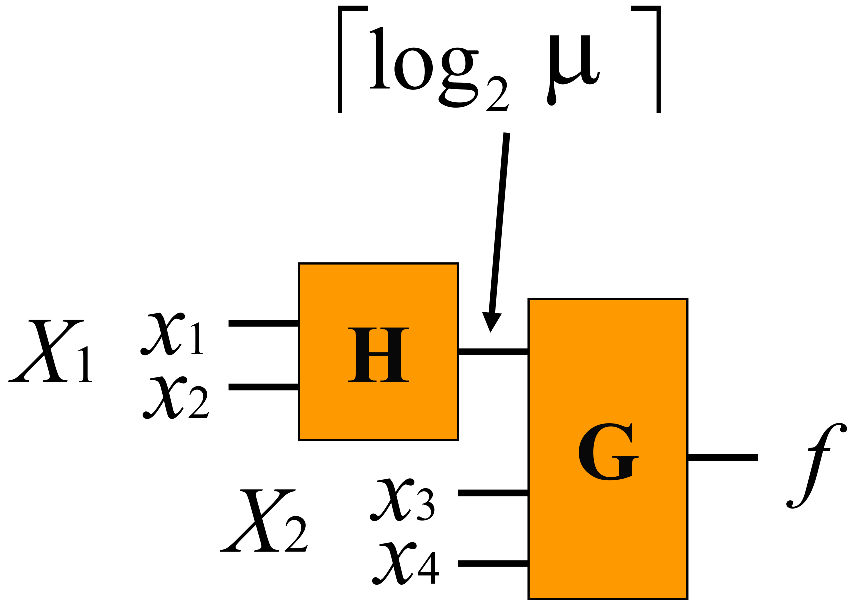
$X_1=(x_1, x_2)$

Free Variables

$X_2=(x_3, x_4)$

	00	01	10	11
00	**0**	**1**	**0**	**1**
01	**1**	**1**	**1**	**1**
10	**1**	**0**	**1**	**0**
11	**1**	**0**	**1**	**0**
$h(X_1)$	0	1	0	1

Column Multiplicity $\mu=2$



$$f=g(h(X_1),X_2)$$

Example

$X_1=(x_1, x_2)$

$X_2=(x_3, x_4)$

	00	01	10	11
00	**0**	**1**	**0**	**1**
01	**1**	**1**	**1**	**1**
10	**1**	**0**	**1**	**0**
11	**1**	**0**	**1**	**0**
$h(X_1)$	0	1	0	1

$\mu=2$

$2^4 \times 1=16$ [bit]

x1	0	0	1	1
x2	0	1	0	1
h(X1)	0	1	0	1

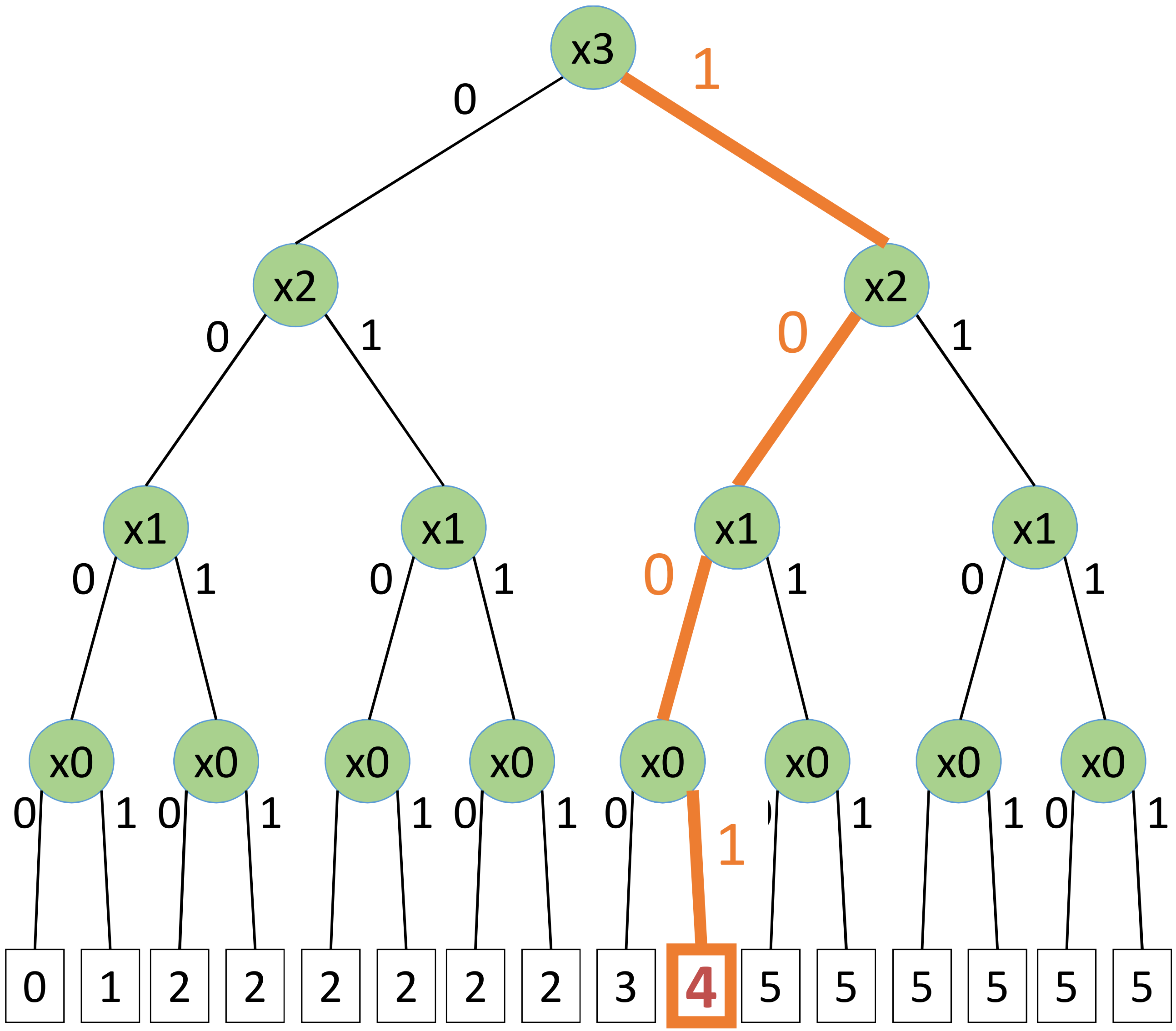
x3,x4	0	1
00	0	1
01	1	1
10	1	0
11	1	0

h(X1)

$2^2 \times 1+2^3 \times 1=12$ [bit]

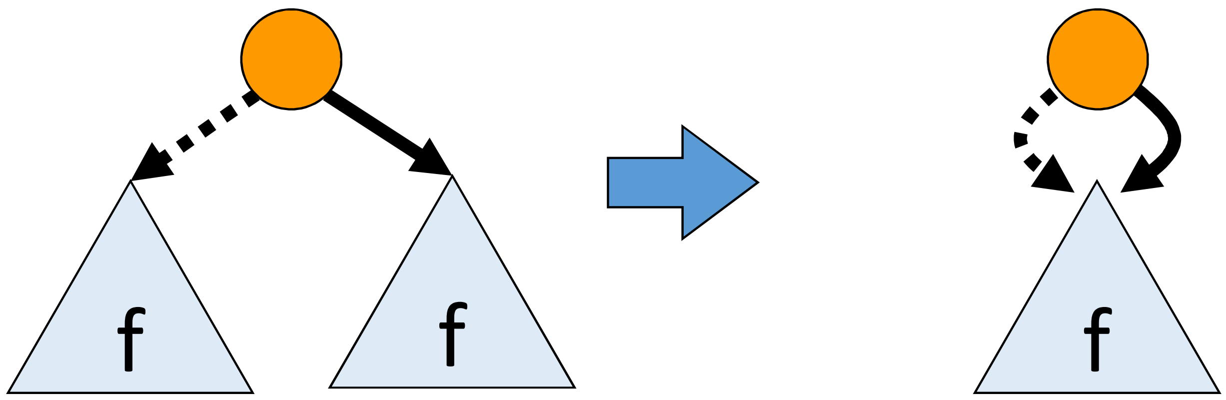
Binary Tree

SA	(x3x2x1x0)	IDX_{SA}
0	(0000)	0
1	(0001)	1
2	(0010)	2
3	(0011)	2
4	(0100)	2
5	(0101)	2
6	(0110)	2
7	(0111)	2
8	(1000)	3
9	(1001)	4
10	(1010)	5
11	(1011)	5
12	(1100)	5
13	(1101)	5
14	(1110)	5
15	(1111)	5

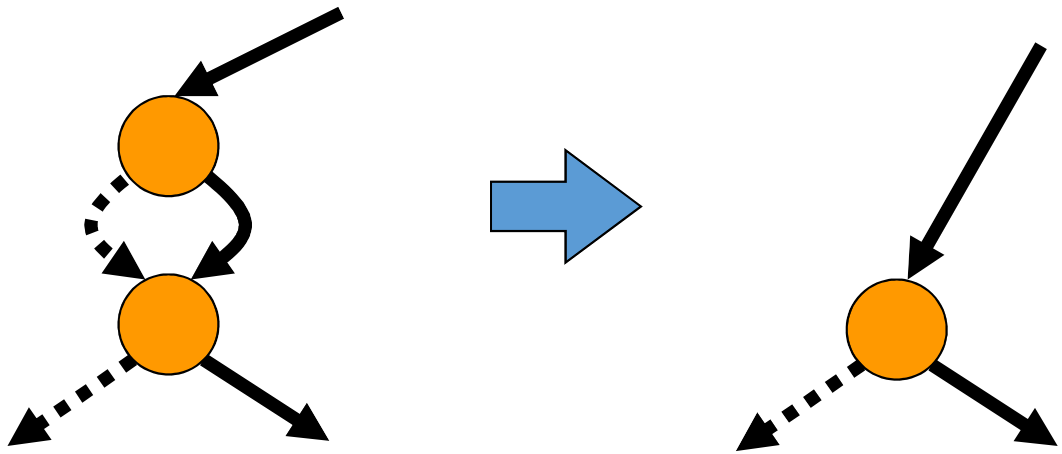


Reduction Rules of a Binary Decision Diagram (BDD)

Rule 1: Merge

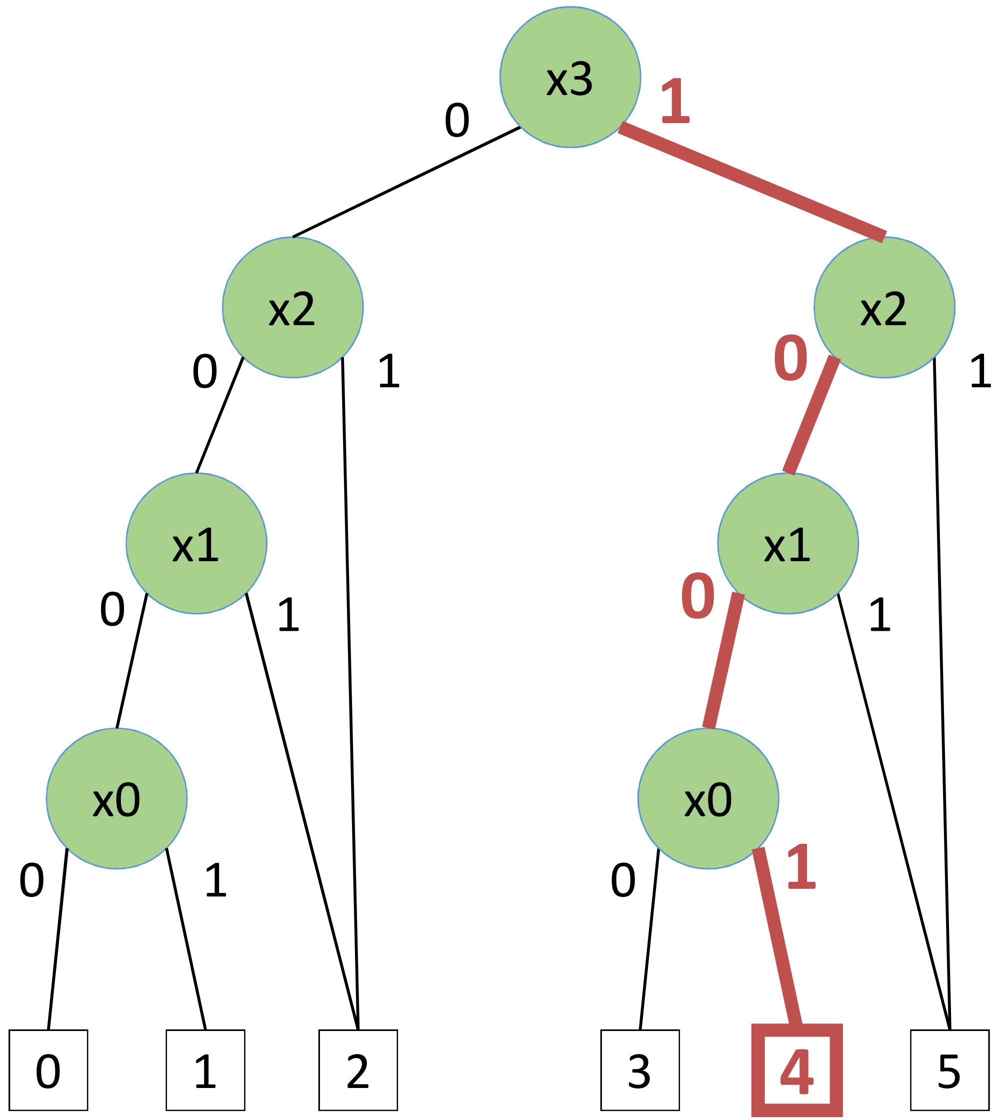


Rule 2: Delete



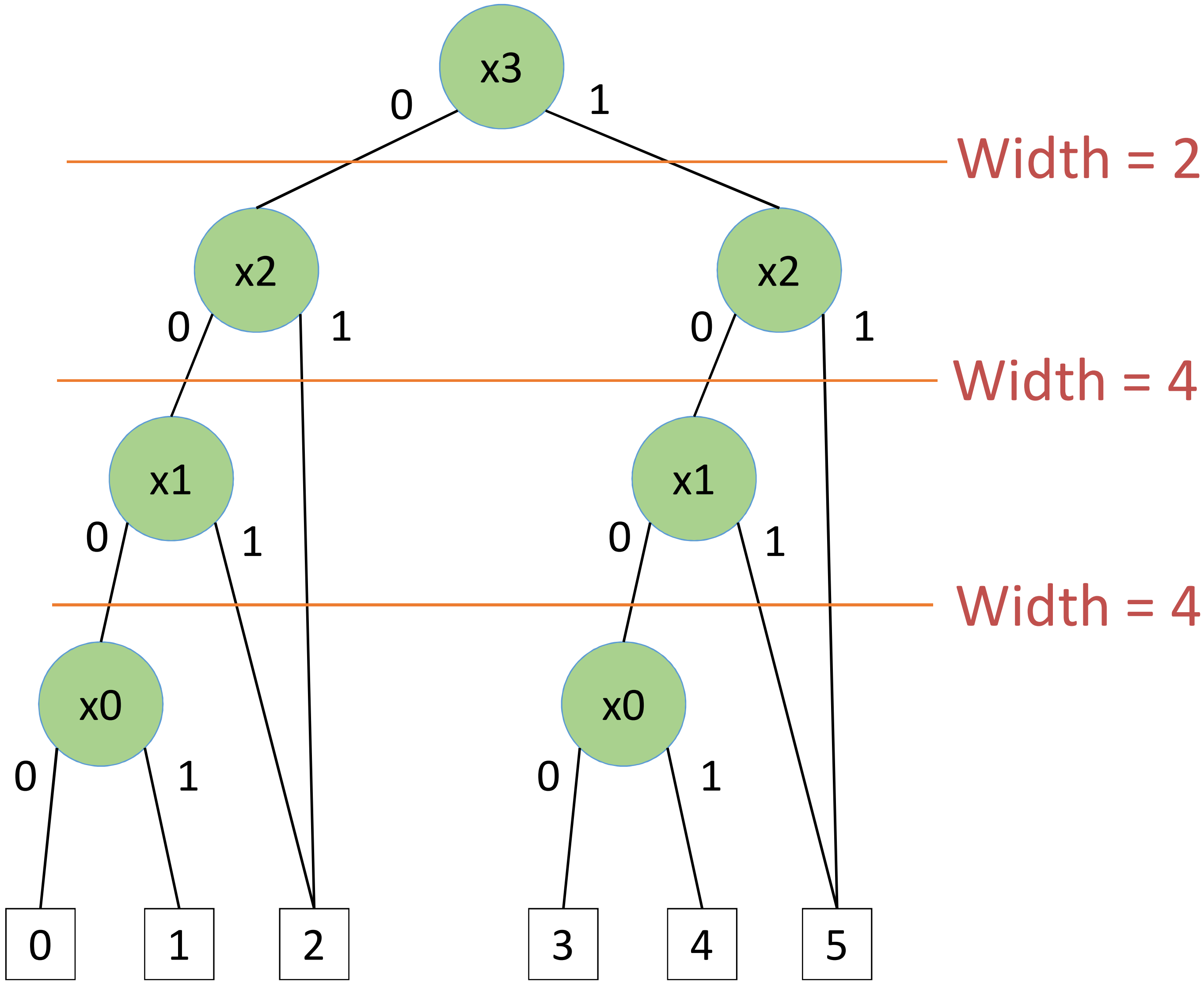
Binary Decision Diagram (BDD)

SA	(x3x2x1x0)	IDX_{SA}
0	(0000)	0
1	(0001)	1
2	(0010)	2
3	(0011)	2
4	(0100)	2
5	(0101)	2
6	(0110)	2
7	(0111)	2
8	(1000)	3
9	(1001)	4
10	(1010)	5
11	(1011)	5
12	(1100)	5
13	(1101)	5
14	(1110)	5
15	(1111)	5



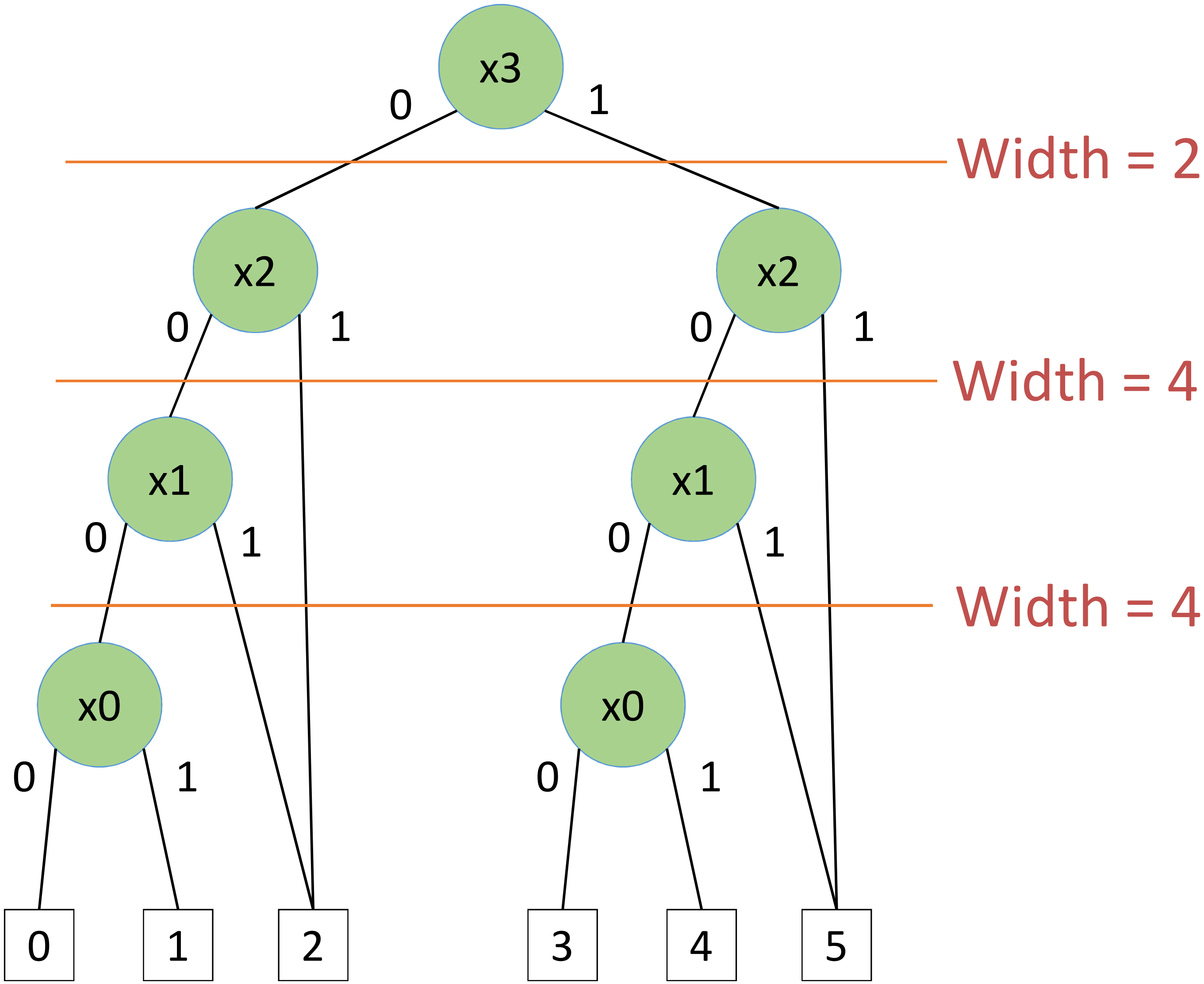
Width

- # of edges crossing the section of the BDD between x_k and x_{k+1}

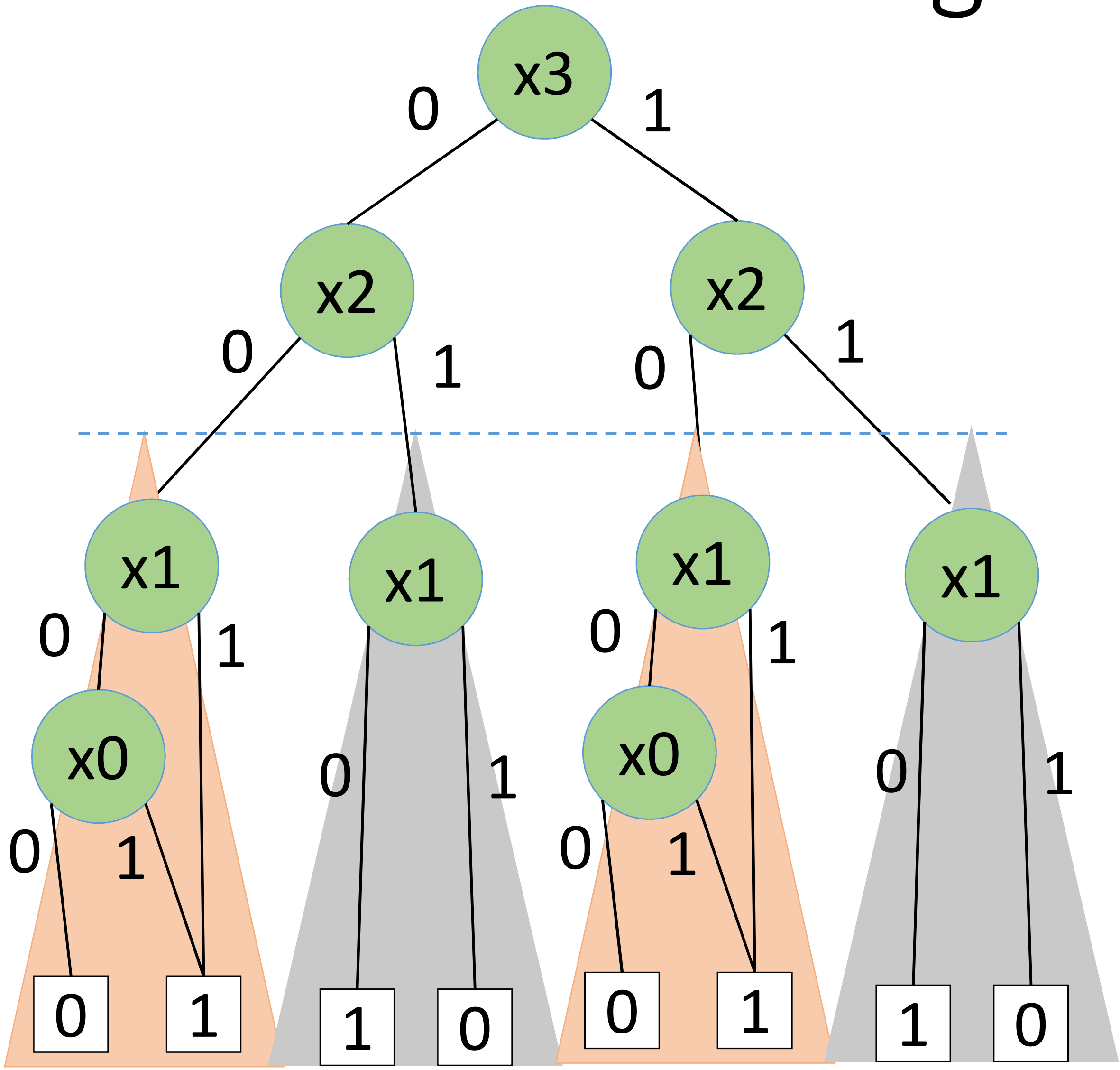


C-measure for a BDD

- Maximum # of width for the BDD



Functional Decomposition using the Decision Diagram

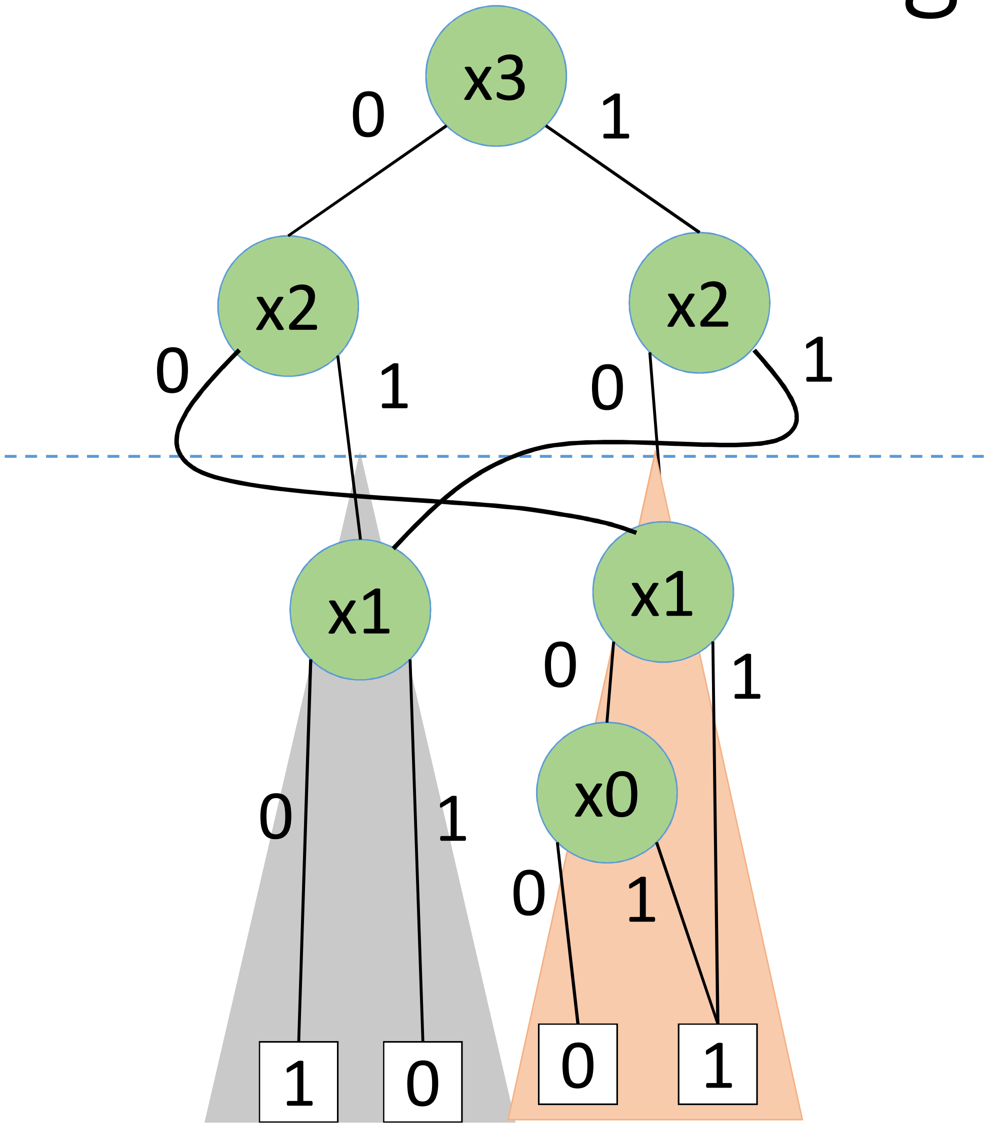


x1	0	0	1	1
x2	0	1	0	1
h(X1)	0	1	0	1

h(X1)

x3, x4	0	1	0	1
00	0	1	0	1
01	1	1	1	1
10	1	0	1	0
11	1	0	1	0

Functional Decomposition using the Decision Diagram



x1	0	0	1	1
x2	0	1	0	1
h(X1)	0	1	0	1

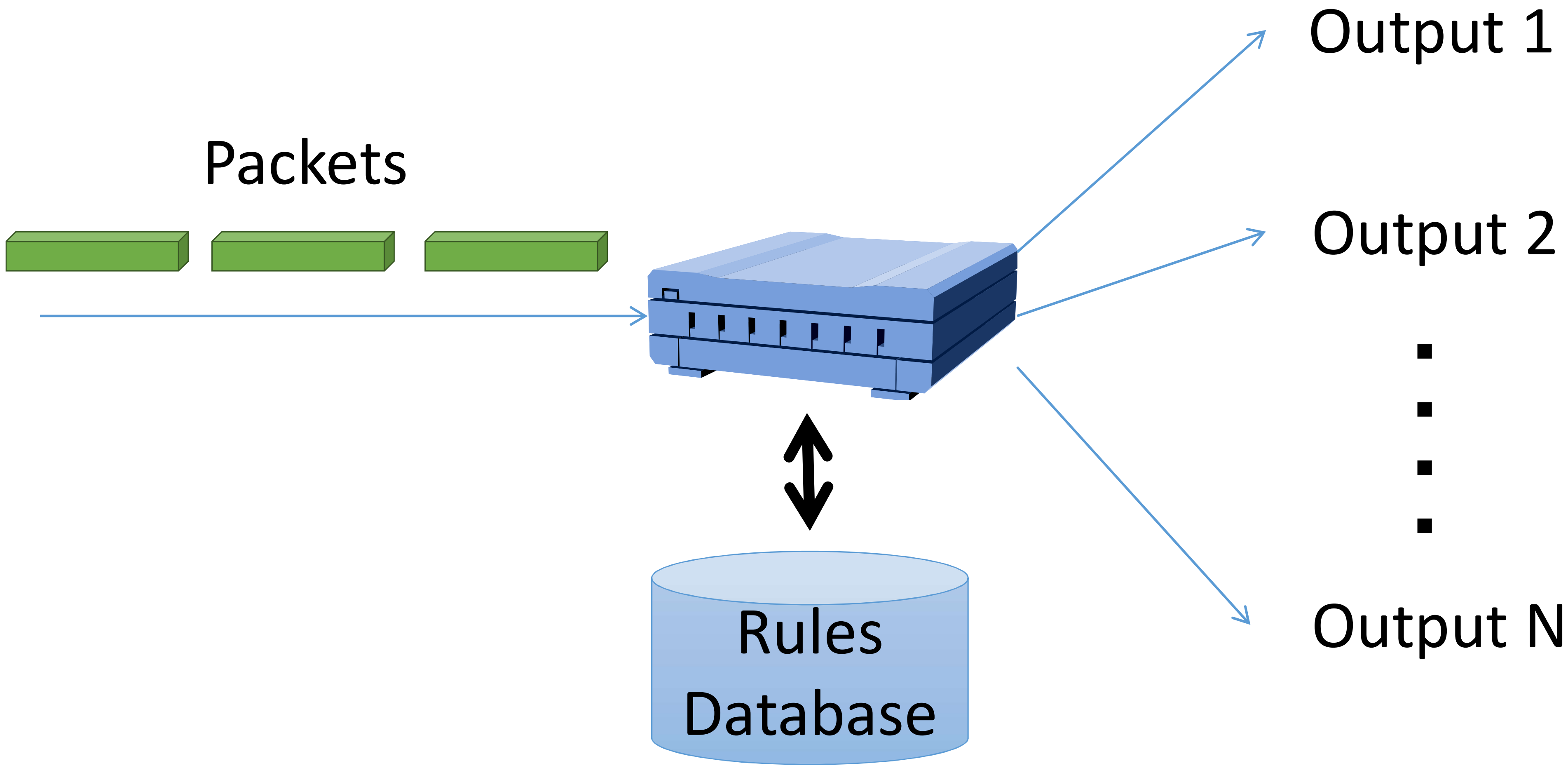
h(X1)

		0	1
x3,x4	00	0	1
	01	1	1
	10	1	0
	11	1	0

3. Case studies

3.1. Index Generation Function

Packet Classification on a Router



Pattern Matching

0 0 0 1

x_1	x_2	x_3	x_4	f
0	0	0	1	1
0	0	1	1	2
0	1	0	1	3
0	1	1	0	4
0	0	0	0	5
0	1	1	1	6
1	0	0	0	7

Match

5-tuple Packet Classification Problem

- Input：Packet Header （104 bit）
- Output：Matched Rule Number

32 bit	16 bit	32 bit	16 bit	8 bit
SA	SP	DA	DP	PRT

Source Adr	Source Port	Dest. Adr	Dest. Port	Proto	Rule
10.0.0.1/4	0:80	131.26.2.1/9	*	TCP	3
192.1.0.1/8	1024	20.1.1.3/24	16:32	UDP	2
255.8.0.0/2	*	254.3.1.0/16	*	ICMP	1
*	*	*	*	*	0

Terminology

Entry

Source Adr	Source Port	Dest. Adr	Dest. Port	Proto	Rule
10.0.0.1/4	0:80	131.26.2.1/9	*	TCP	3
192.1.0.1/8	1024	20.1.1.3/24	16:32	UDP	2
255.8.0.0/2	*	254.3.1.0/16	*	ICMP	1
*	*	*	*	*	0

Field

Rule

Example of 3-Field Packet Classification Table

SA	SP	PRT	Rule
1000	1000:1001	0010	3
01**	0110:1000	0001	2
010*	0111:1110	****	1
****	0000:1111	****	0

Longest Prefix Matching

Range Matching

Exact Matching

Interval Function

- Input: $x_i \in \{0,1\}$, $X=(x_1, x_2, \ldots, x_n)$, $Y = \sum_{i=0}^{n-1} x_{i+1} \times 2^i$
- Integers: A, B, $0 \leq A \leq B \leq 2^n - 1$

$$IN(X{:}A,B) = \begin{cases} 1 & \text{if } A \leq Y \leq B \\ 0 & \text{otherwise} \end{cases}$$

Representation by Interval Functions

SA	SP	PRT	Rule
1000	1000:1001	0010	3
01**	0110:1000	0001	2
010*	0111:1110	****	1
****	0000:1111	****	0

SA	SP	PRT	Rule
$IN(X_{SA},8,8)$	$IN(X_{SP},8,9)$	$IN(X_{PRT},2,2)$	3
$IN(X_{SA},4,7)$	$IN(X_{SP},6,8)$	$IN(X_{PRT},1,1)$	2
$IN(X_{SA},4,5)$	$IN(X_{SP},7,14)$	$IN(X_{PRT},0,15)$	1
$IN(X_{SA},0,15)$	$IN(X_{SP},0,15)$	$IN(X_{PRT},0,15)$	0

Decomposed Realization

SA	IDX_{SA}
$IN(X_{SA},0,3)$	0
$IN(X_{SA},4,5)$	1
$IN(X_{SA},6,7)$	2
$IN(X_{SA},8,8)$	3
$IN(X_{SA},9,15)$	4

SP	IDX_{SP}
$IN(X_{SP},0,0)$	0
$IN(X_{SP},1,1)$	1
$IN(X_{SP},2,7)$	2
$IN(X_{SP},8,8)$	3
$IN(X_{SP},9,9)$	4
$IN(X_{SP},10,15)$	5

PRT	IDX_{PRT}
$IN(X_{PRT},0,0)$	0
$IN(X_{PRT},2,2)$	1
$IN(X_{PRT},3,15)$	2

Field Functions

IDX_{SA}	IDX_{SP}	IDX_{PRT}	Rule
4	4	3	1
4	5	3	1
2	2	3	2
⋮	⋮	⋮	⋮

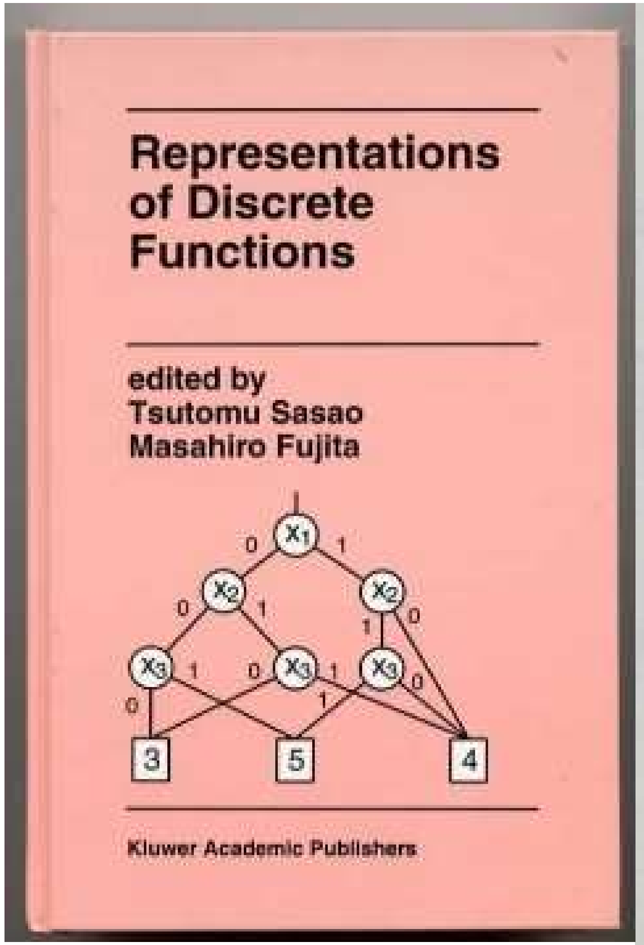
Cartesian Product Function

Example of Field Function

SA	IDX_{SA}
$IN(X_{SP},0,0)$	0
$IN(X_{SP},1,1)$	1
$IN(X_{SP},2,7)$	2
$IN(X_{SP},8,8)$	3
$IN(X_{SP},9,9)$	4
$IN(X_{SP},10,15)$	5

SA	IDX_{SA}
0	0
1	1
2	2
3	2
4	2
5	2
6	2
7	2
8	3
9	4
10	5
11	5
12	5
13	5
14	5
15	5

Further Reduction of Width by an Edge-valued BDD (EVBDD)



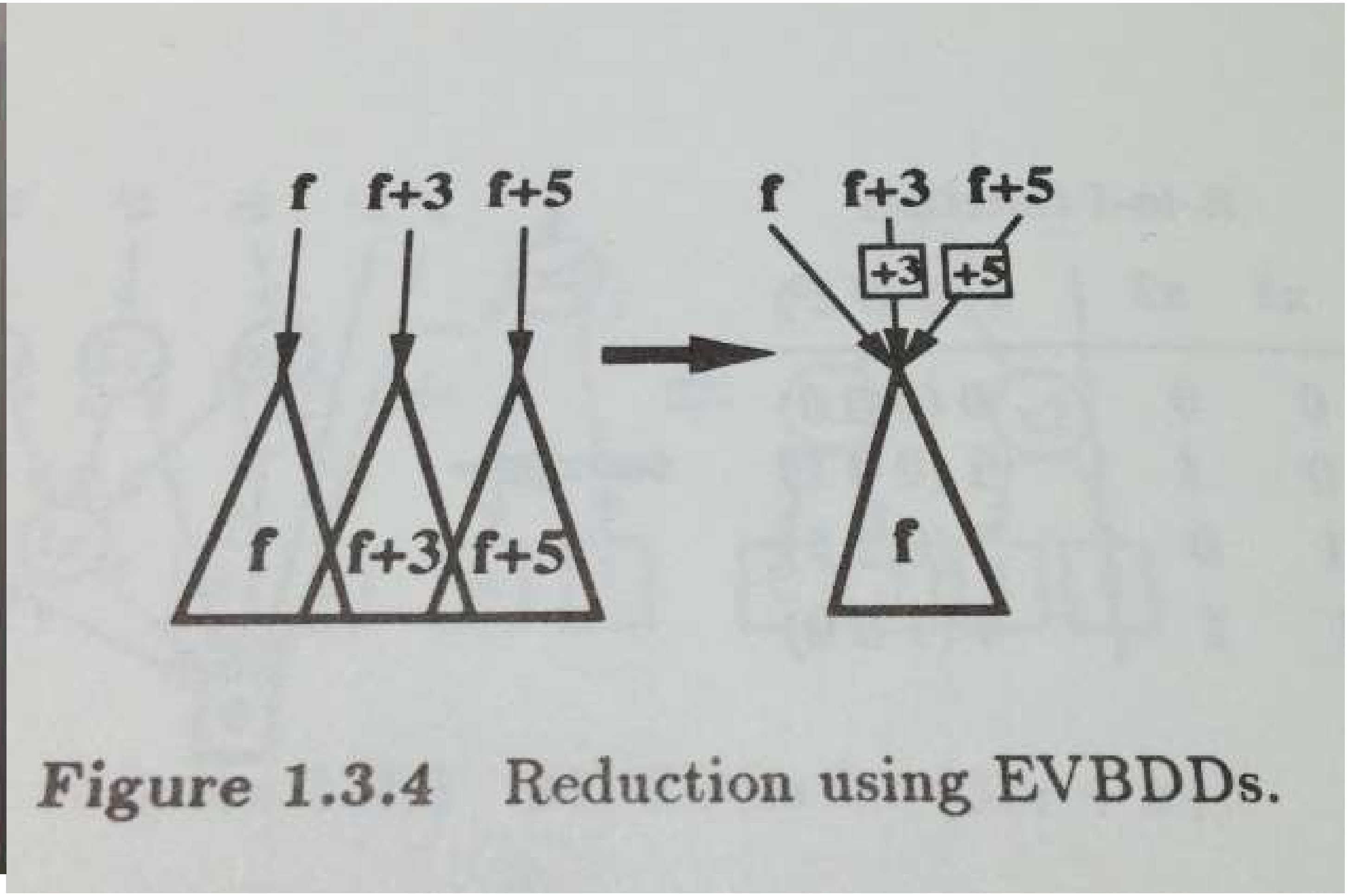
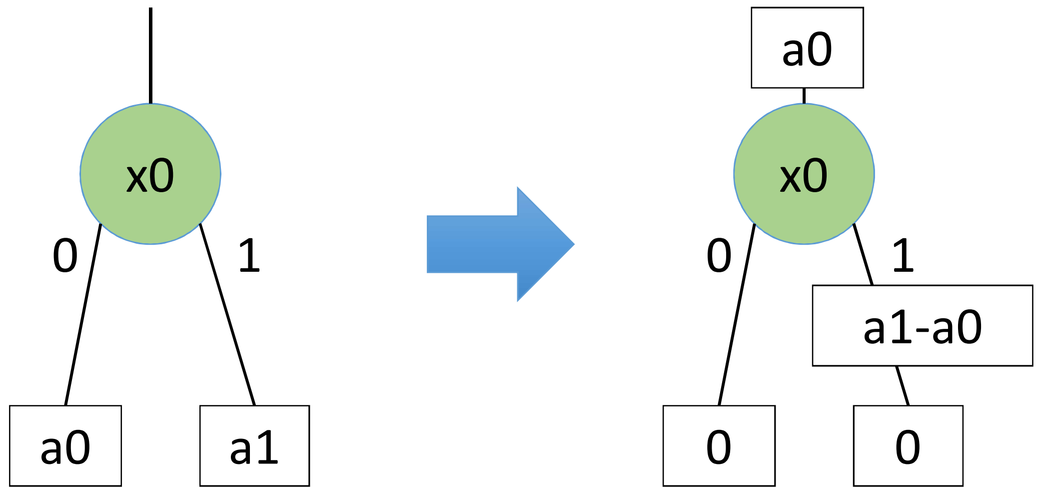


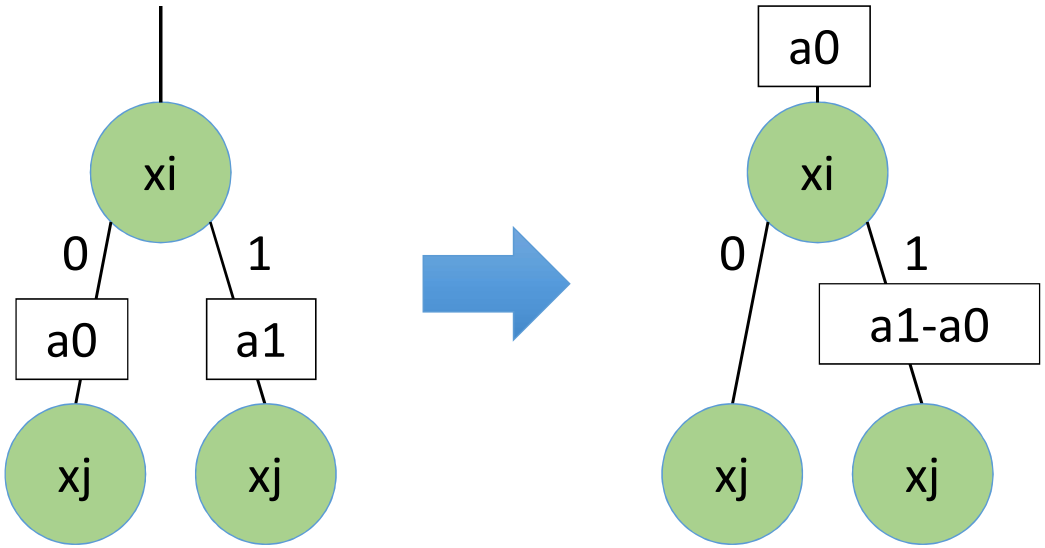
Figure 1.3.4 Reduction using EVBDDs.

BDD to EVBDD Conversion Rules

Rule 1: Terminal node

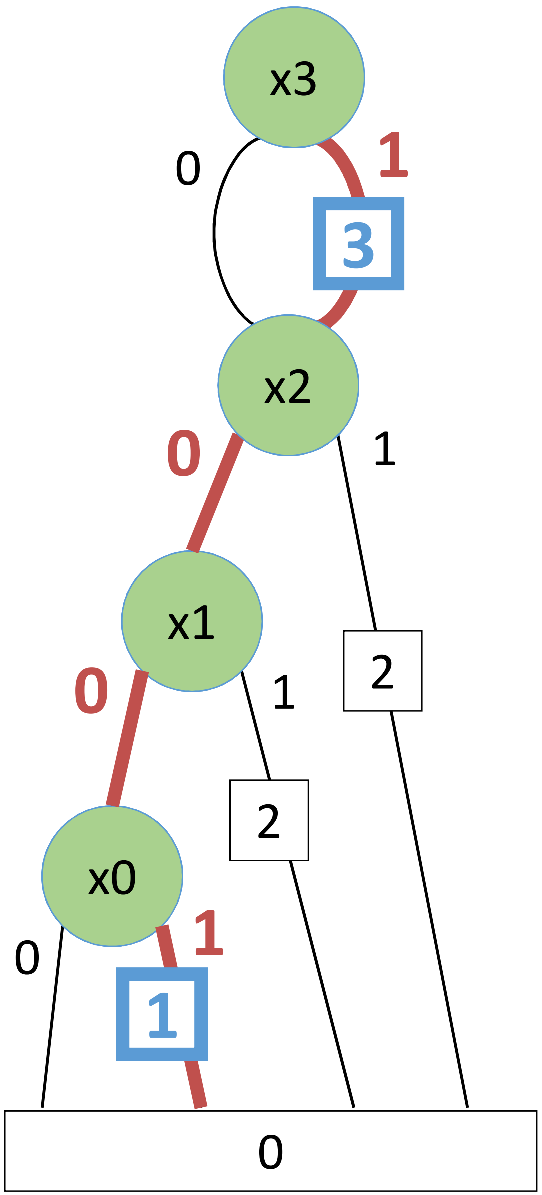


Rule 2: Non-terminal node

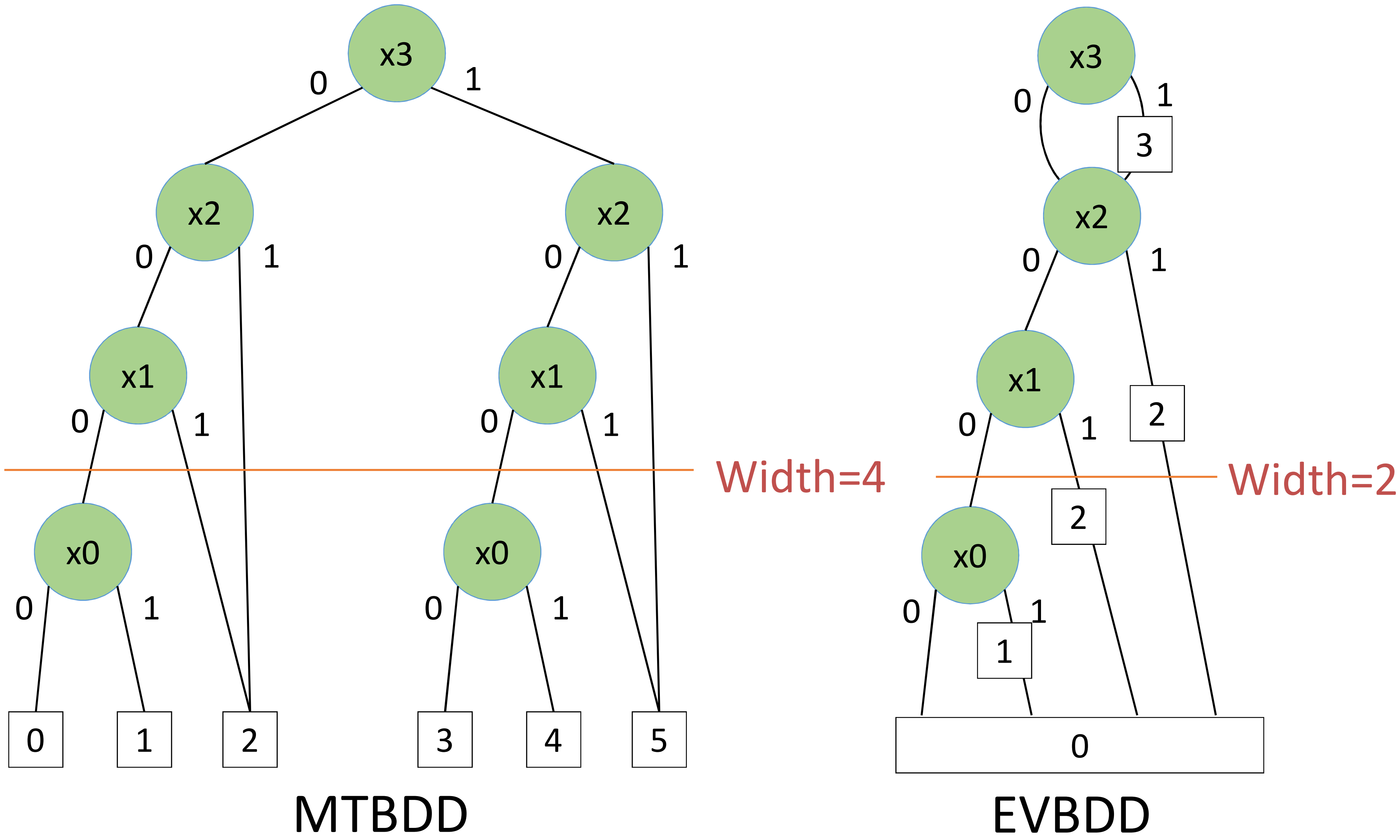


Example of EVBDD

SA	(x3x2x1x0)	IDX_{SA}
0	(0000)	0
1	(0001)	1
2	(0010)	2
3	(0011)	2
4	(0100)	2
5	(0101)	2
6	(0110)	2
7	(0111)	2
8	(1000)	3
9	(1001)	4
10	(1010)	5
11	(1011)	5
12	(1100)	5
13	(1101)	5
14	(1110)	5
15	(1111)	5



MTBDD vs. EVBDD



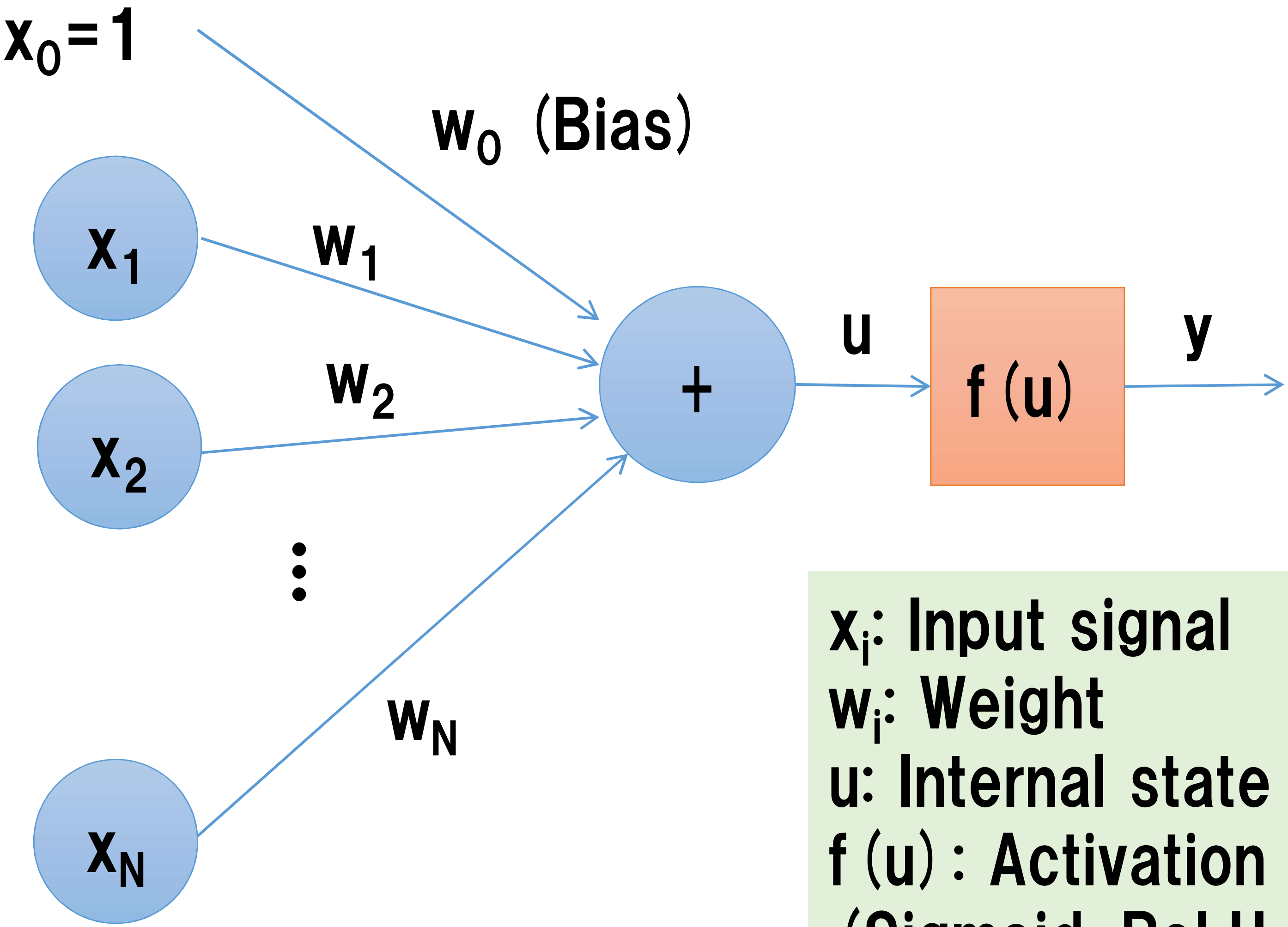
Theorems

- Theorem 3.1.1: A field function with p distinct intervals has at most 2p+1 column multiplicities.

- Theorem 3.1.2: Let t be the number of unique indices for the monotone-increasing function. Then, there exist an EVBDD with at most t column multiplicities.

3.2. Weight-Sum Function

Artificial Neuron (AN)



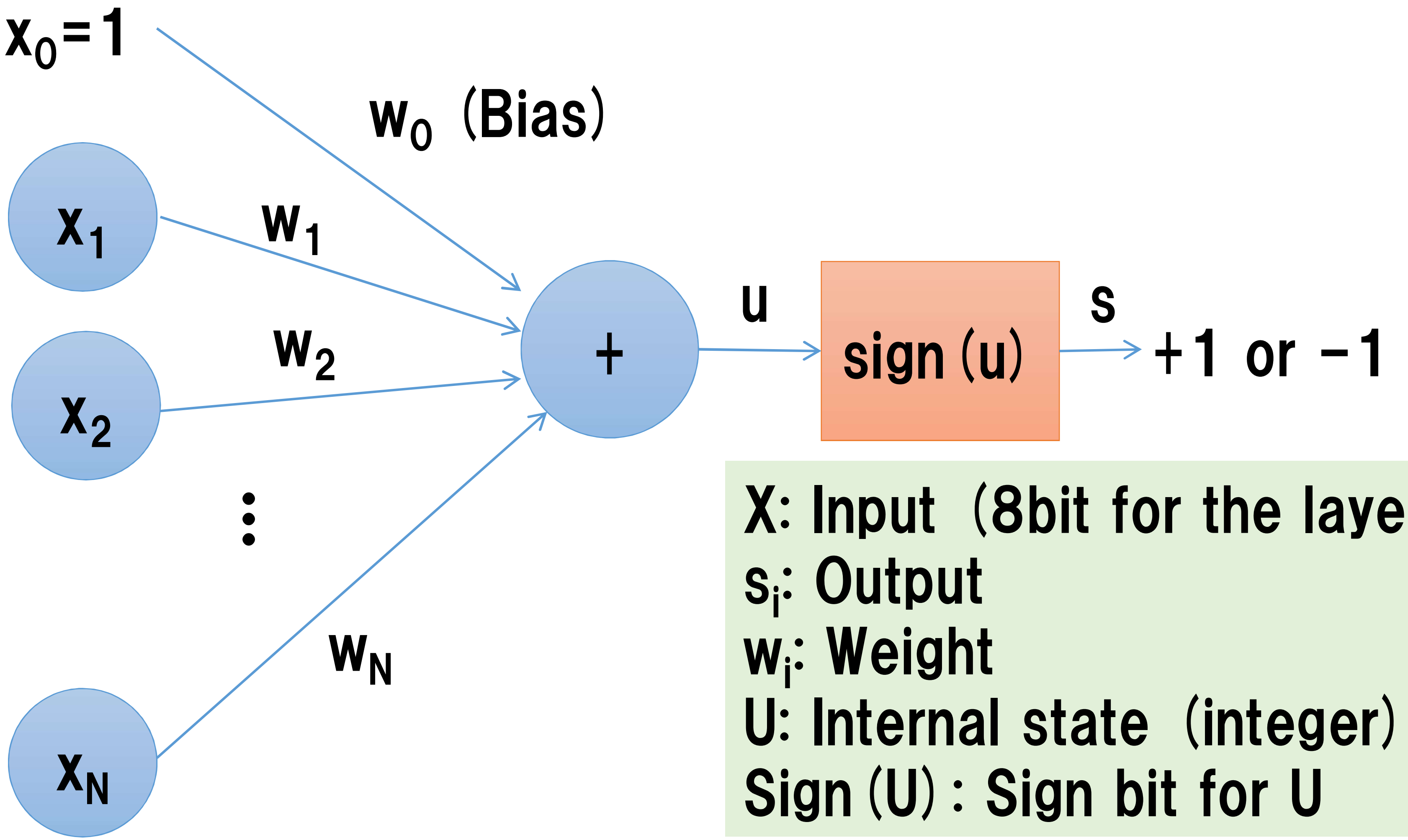
$$y = f(u)$$

$$u = \sum_{i=0}^{N} w_i x_i$$

x_i: Input signal
w_i: Weight
u: Internal state
f (u) : Activation function
(Sigmoid, ReLU, etc.)
y: Output signal

Binarized DCNN

- Treats only binarized (+1/-1) values (weights and inouts)
    - Except for the first and the last layers



X: Input (8bit for the layer 1)
s_i: Output
w_i: Weight
U: Internal state (integer)
Sign (U) : Sign bit for U

H. Nakahara, H. Yonekawa, H. Iwamoto and M. Motomura, “A memory-based realization of a Binarized deep convolutional neural network,” *FPT2016* (accepted).

Example of Sum of Binarized-weights (n=5)

s_0	s_1	s_2	s_3	s_4	f(S)
0	0	0	0	0	$-w_0-w_1-w_2-w_3-w_4$
0	0	0	0	1	$-w_0-w_1-w_2-w_3+w_4$
0	0	0	1	0	$-w_0-w_1-w_2+w_3-w_4$
0	0	0	1	1	$-w_0-w_1-w_2-w_3+w_4$
0	0	1	0	0	$-w_0-w_1+w_2-w_3-w_4$
0	0	1	0	1	$-w_0-w_1+w_2-w_3+w_4$
0	0	1	1	0	$-w_0-w_1+w_2+w_3-w_4$
0	0	1	1	1	$-w_0-w_1+w_2+w_3+w_4$
0	1	0	0	0	$-w_0+w_1-w_2-w_3-w_4$
0	1	0	0	1	$-w_0+w_1-w_2-w_3+w_4$
0	1	0	1	0	$-w_0+w_1-w_2+w_3-w_4$
0	1	0	1	1	$-w_0+w_1-w_2+w_3+w_4$
		⋮			⋮
1	1	1	1	1	$+w_0+w_1+w_2+w_3+w_4$

Analysis of a weight-sum function

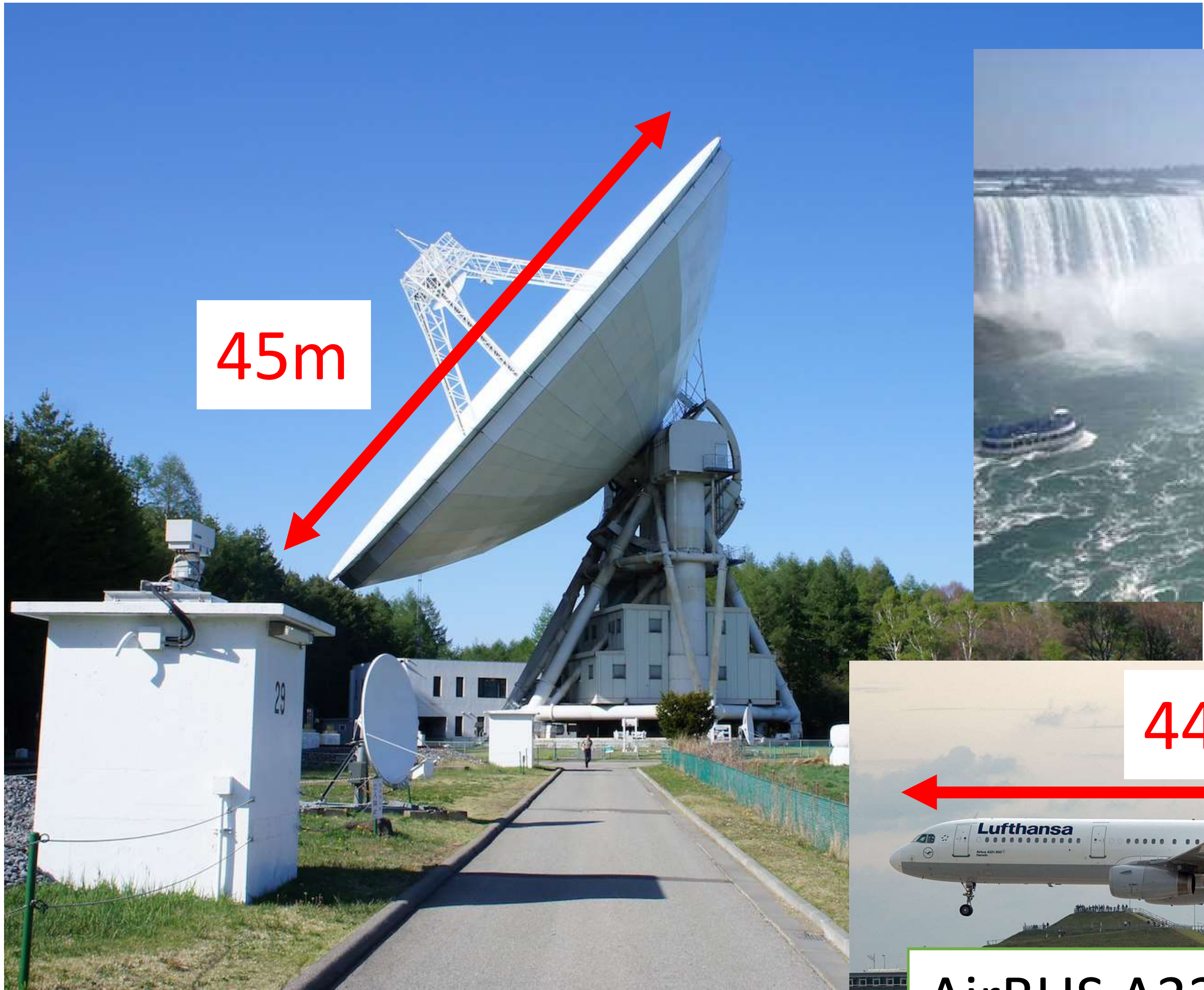
$$u = \sum_{i=0}^{N} w_i x_i$$

- Theorem 1: When the output of the WS function is represented by q-bits, its column multiplicity is at most 2^q
- Corollary 1: When the output of the WS function is represented by q-bits, its number of rails for the LUT cascade is at most q

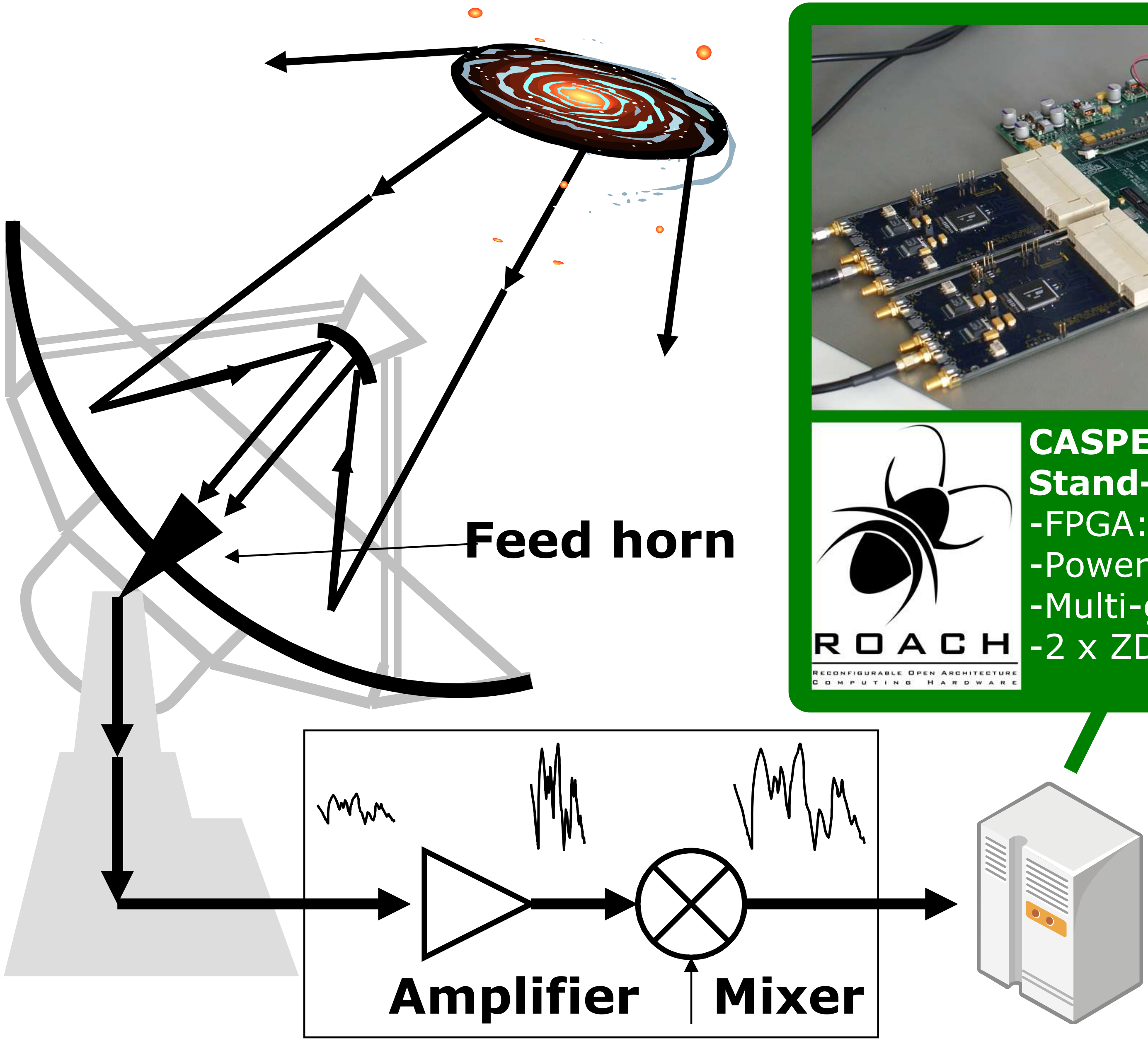
3.3. Residue Number System (RNS)

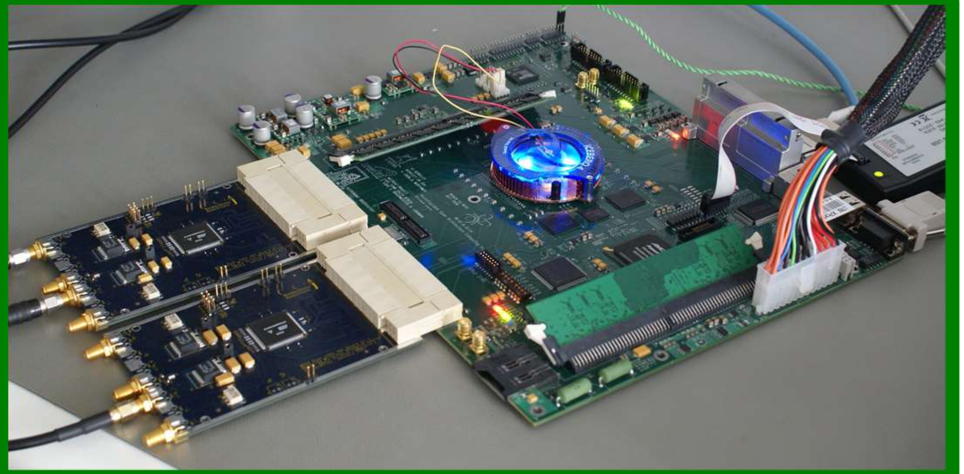
Radio telescope

Radio telescope

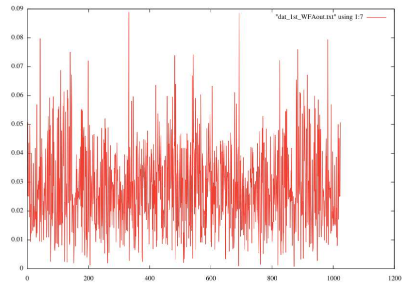


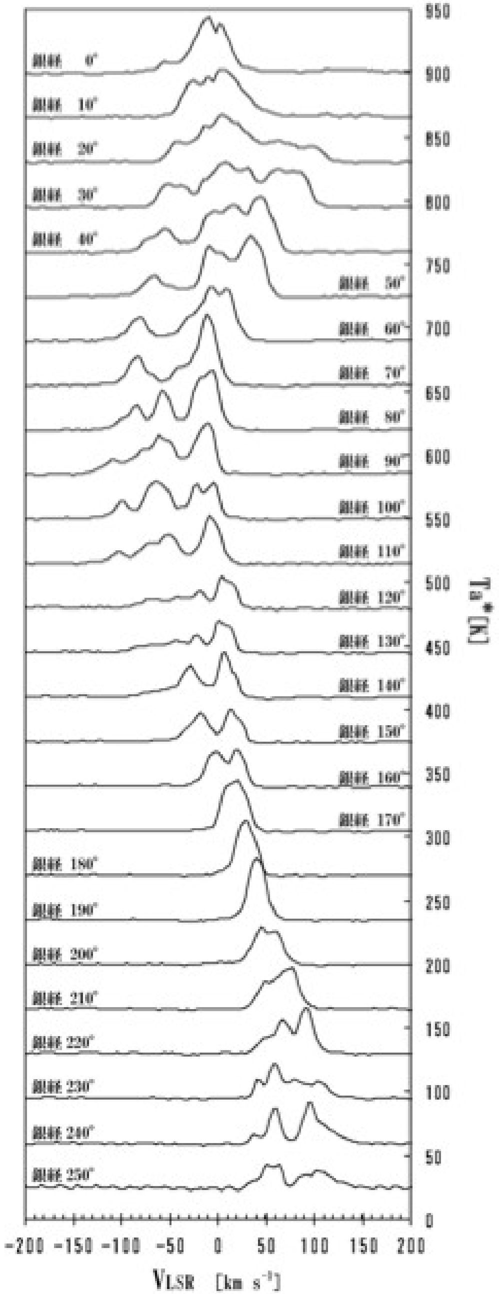
FFT for a Radio Telescope



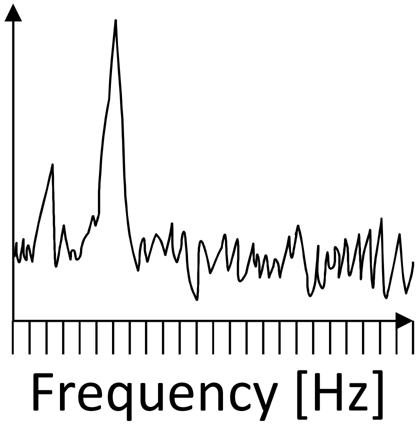


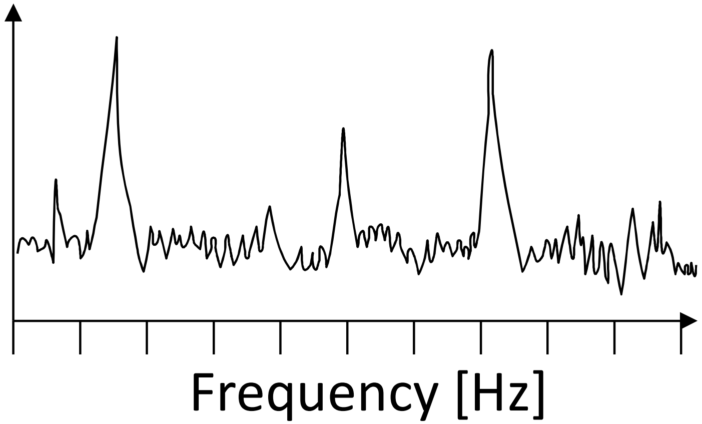
CASPER ROACH-2 Revision 2
Stand-alone FPGA board
- FPGA: Xilinx Virtex-6 SX475T
- PowerPC 440 EPx
- Multi-gigabit transceiver (SFP+)
- 2 x ZDOKs





Requirements





High-resolution

2^{30}-2^{40} points FFT

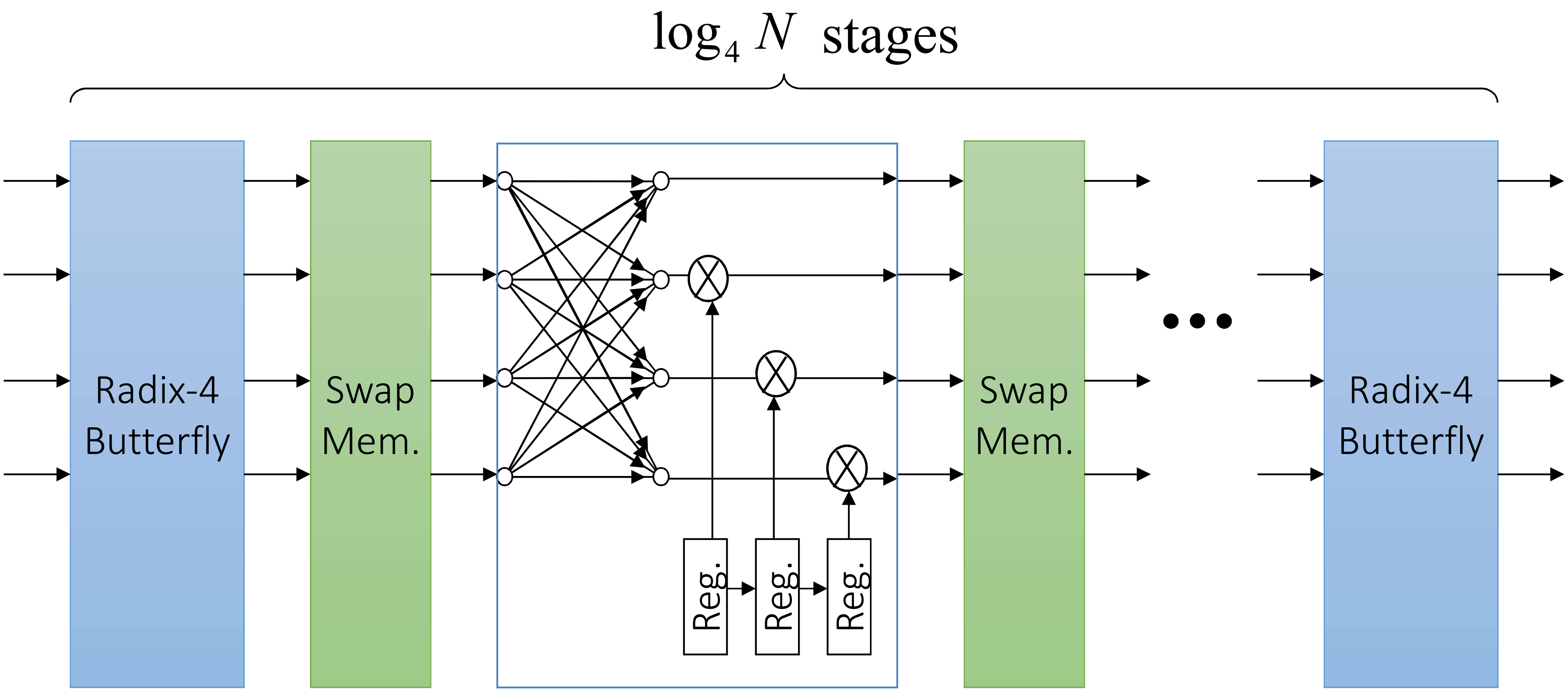
- OFDM: 2^{8}
- CT Scanner: 2^{16}

Wide-band

0.1 – 1000GHz

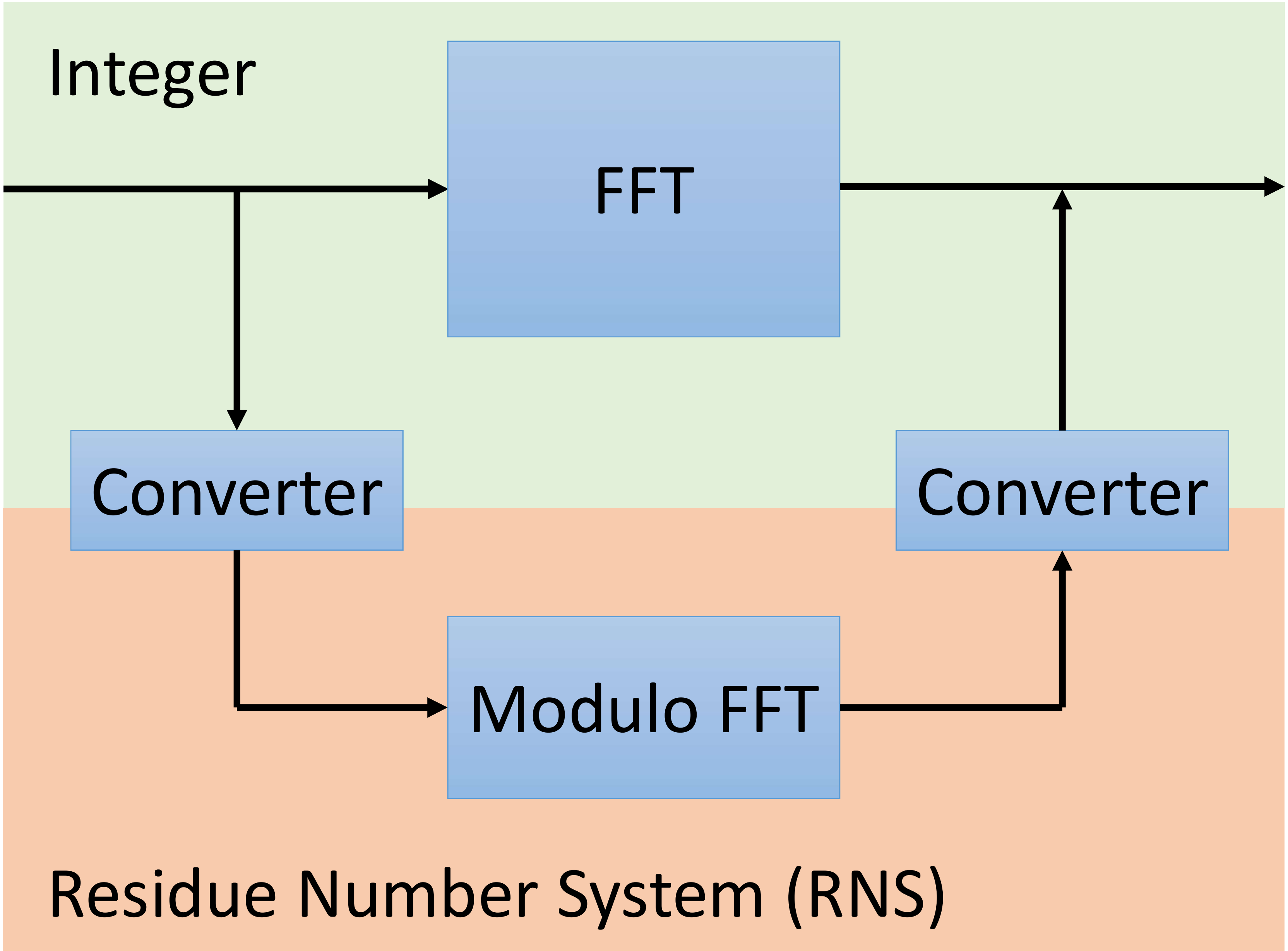
- Digital TV: 470-770MHz (UHF in Japan)
- Cellular phone: 0.8-2GHz

Pipelined Binary FFT



H. Nakahara, H. Nakanishi, and T. Sasao, "On a wideband fast Fourier transform for a radio telescope," *ACM SIGARCH Computer Architecture News*, Vol.40, No. 5, 2012, pp.46-51.

Different Number System



Residue Number System (RNS)

- Defined by a set of L mutually prime integer constants $\langle m_1, m_2, \ldots, m_L \rangle$
- An arbitrary integer X can be represented by a tuple of L integers $(X_1, X_2, \ldots, X_L)$,

  where $X_i \equiv X \pmod{m_i}$
- Dynamic range

$$M = \prod_{i=1}^{L} m_i$$

Multiplication on RNS

- Moduli set〈3,4,5〉, X=8, Y=2
- Z=X×Y=16= (1,0,1)
- X= (2,0,3) , Y= (2,2,2)

Binary2RNS Conversion

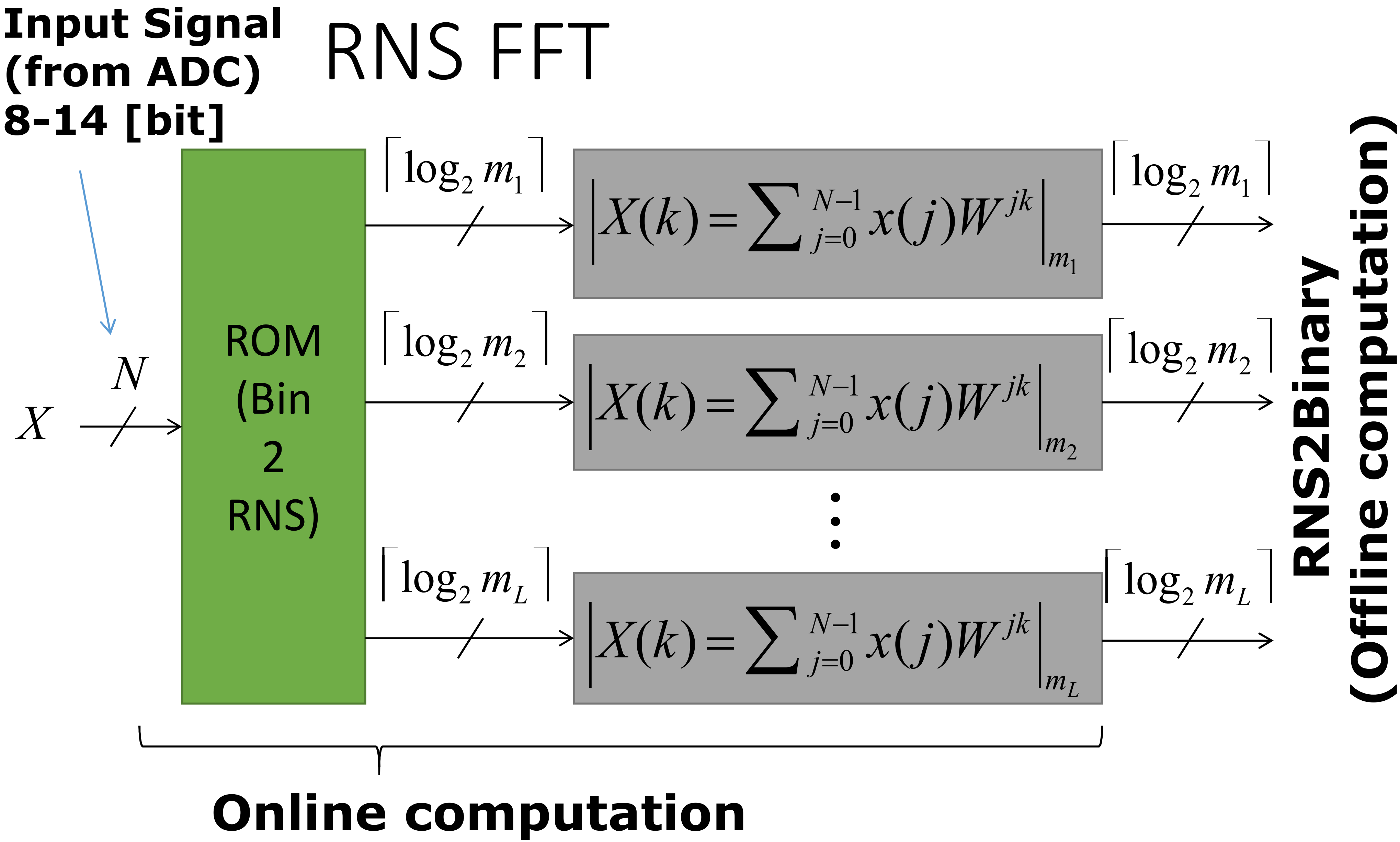
Parallel Multiplication

Z= (4 mod 3,0 mod 4,6 mod 5)

= (1,0,1) =16

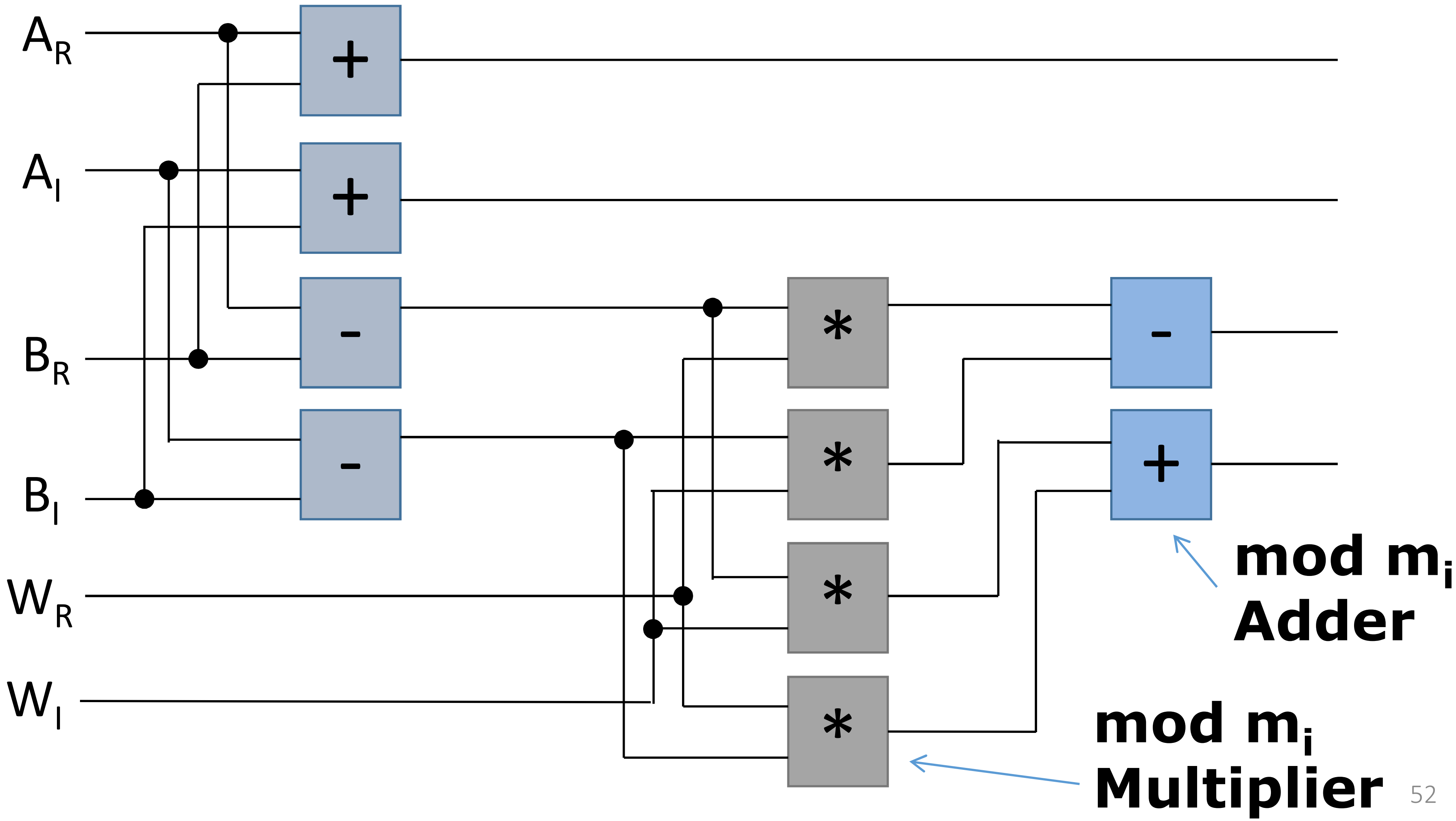
RNS2Binary Conversion

RNS FFT


Input Signal
(from ADC)
8-14 [bit]
X
N
ROM
(Bin
2
RNS)
$\lceil \log_2 m_1 \rceil$
$\lceil \log_2 m_2 \rceil$
$\lceil \log_2 m_L \rceil$
$\left| X(k) = \sum_{j=0}^{N-1} x(j) W^{jk} \right|_{m_1}$
$\left| X(k) = \sum_{j=0}^{N-1} x(j) W^{jk} \right|_{m_2}$
$\left| X(k) = \sum_{j=0}^{N-1} x(j) W^{jk} \right|_{m_L}$
$\lceil \log_2 m_1 \rceil$
$\lceil \log_2 m_2 \rceil$
$\lceil \log_2 m_L \rceil$
RNS2Binary
(Offline computation)
Online computation

Mod m_i RNS Butterfly Operator

R: real part, I: Imaginary part

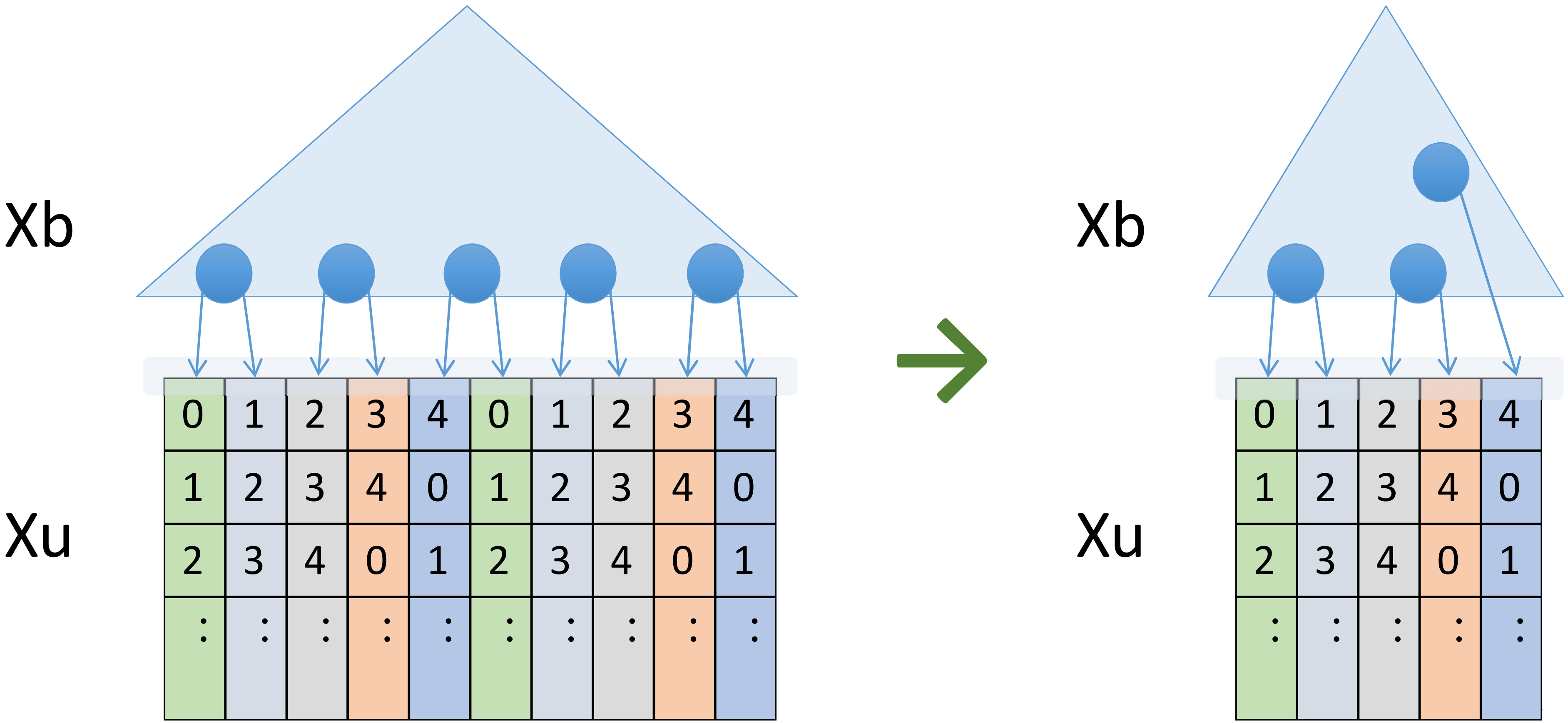


Cyclic Manner in Modulo Operation

1+0 mod 5 = 1 mod 5 = 1
1+1 mod 5 = 2 mod 5 = 2
1+2 mod 5 = 3 mod 5 = 3
1+3 mod 5 = 4 mod 5 = 4
1+4 mod 5 = 5 mod 5 = 0
1+5 mod 5 = 6 mod 5 = 1
1+6 mod 5 = 7 mod 5 = 2
1+7 mod 5 = 8 mod 5 = 3
1+8 mod 5 = 9 mod 5 = 4
1+4 mod 5 = 10 mod 5 = 0
1+5 mod 5 = 11 mod 5 = 1

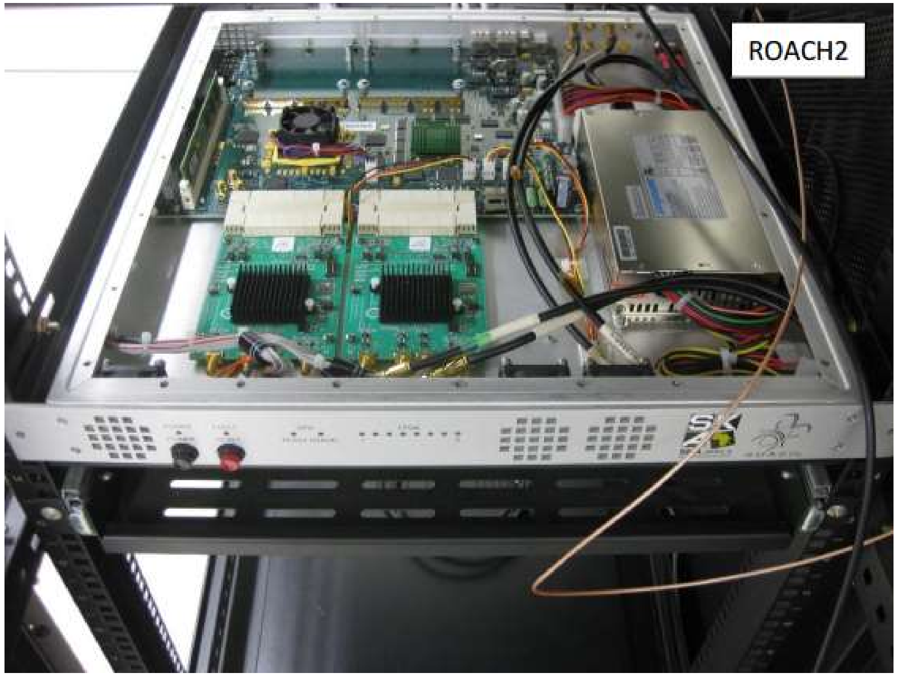
Column Multiplicity for mod m add/sub/mul

- Theorem 2: column multiplicity is at most $\lceil \log_2 m \rceil$
- Ex. m=5



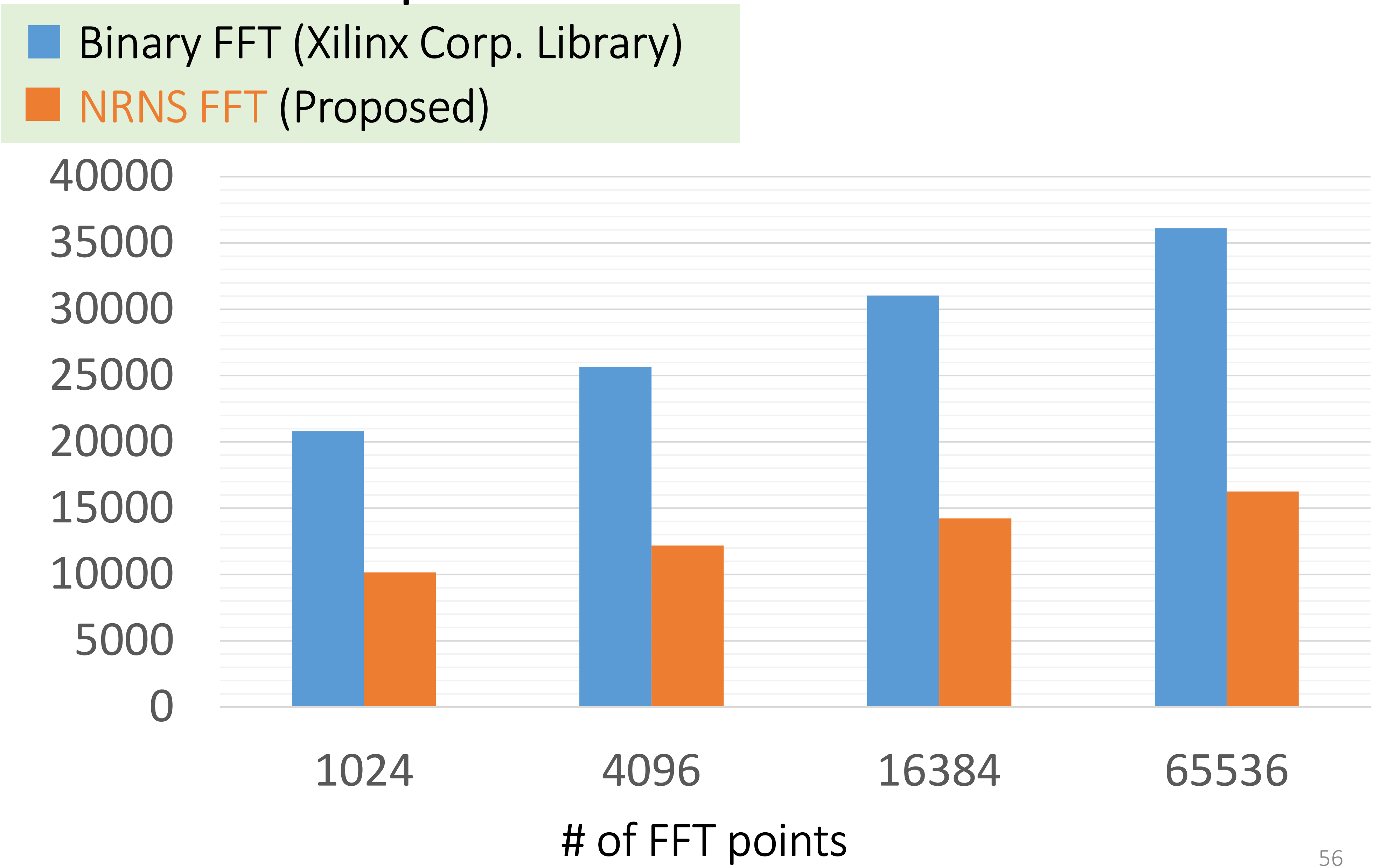
Comparison with Conventional FFTs

- Implemented on the ROACH revision 2 board
  - Xilinx Corp. Virtex6 FPGA
    - 297,600 LUTs
    - 744,000 Slices
    - 2,128 18KbBRAMs
    - 2,016 DSP48Es

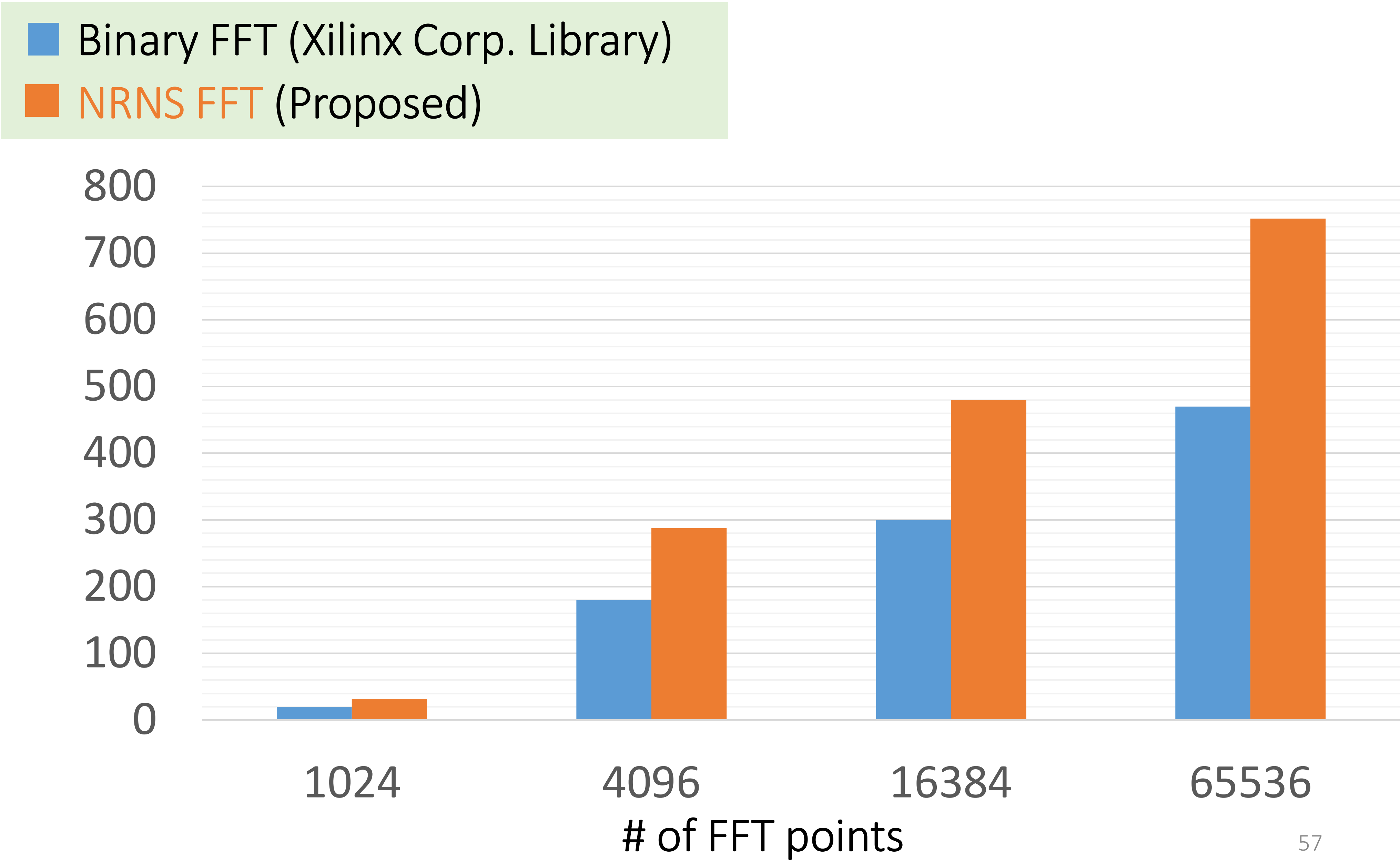


- Compared with the Xilinx FFT Library (version 7.1)
  - CLB logic as a complex multiplexer

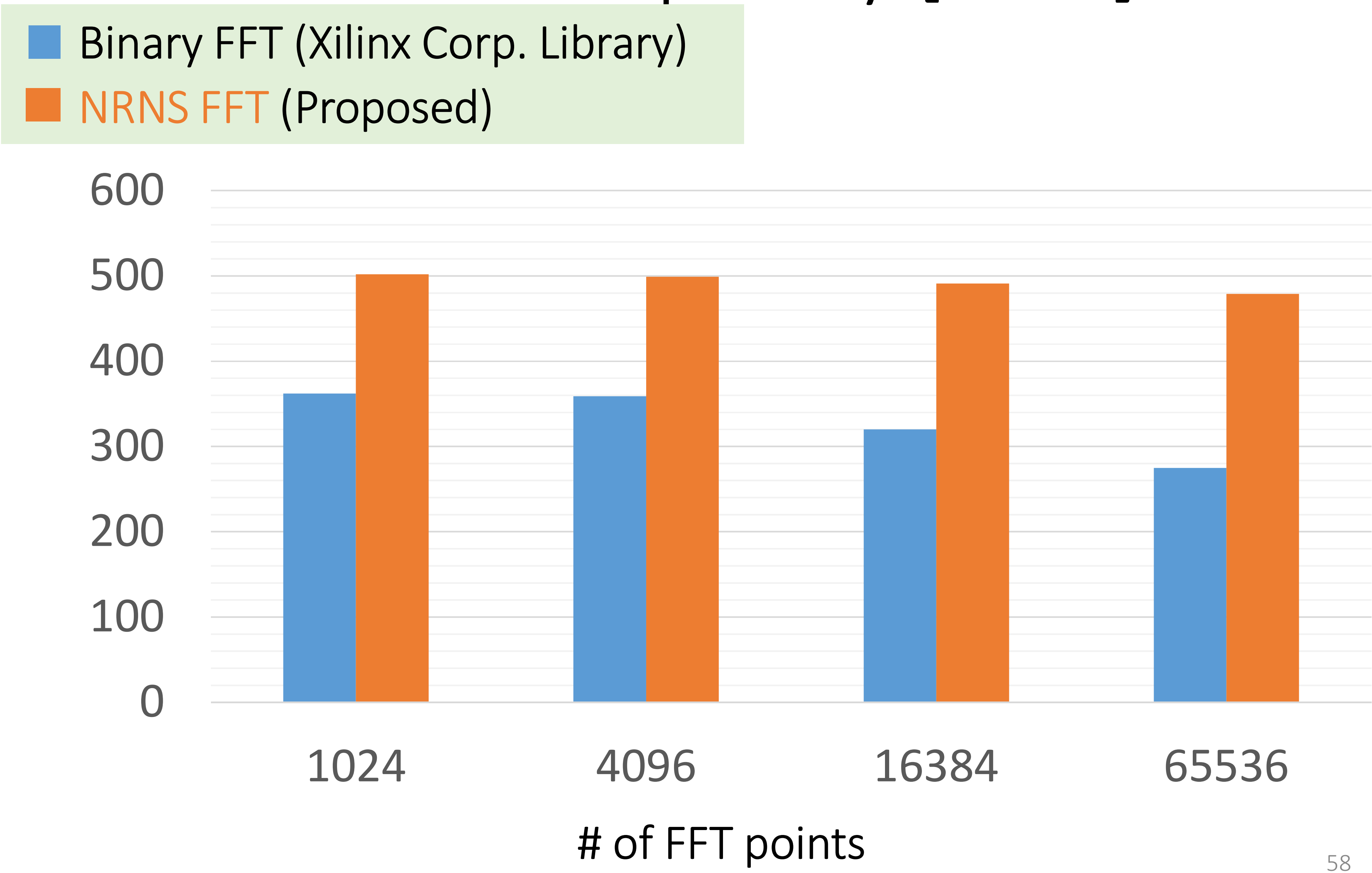
of 6-input LUTs



of 18Kb BRAMs



Max. Clock Frequency [MHz]

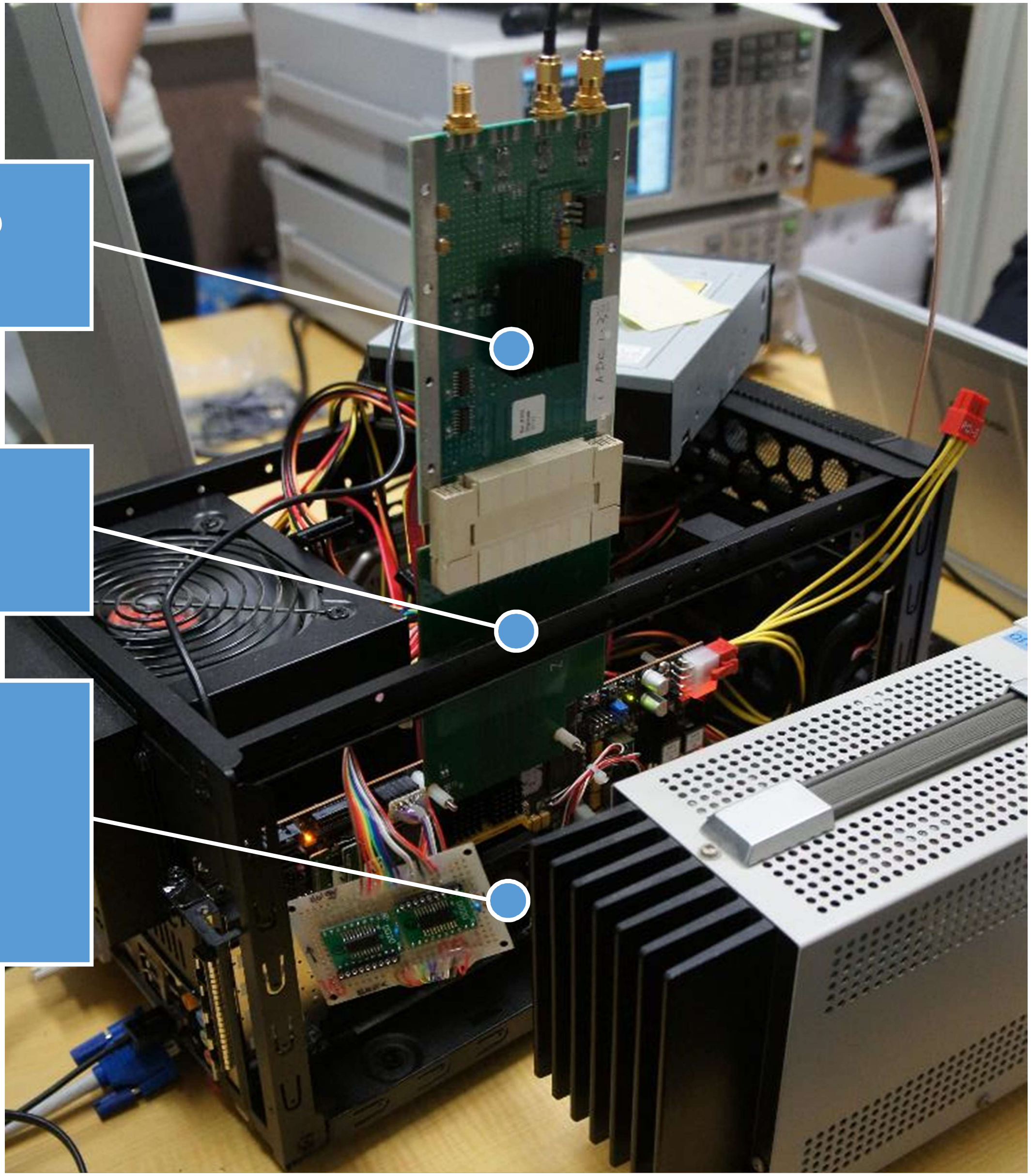


Next ROACH?

CASPER ADC1.3
(5Gsps)

FMC-ZDOC
Connector

Net FPGA SUME
(Virtex7 FPGA)



Summary

- Typical FPGA consists of various memories, such as a look-up table (LUT) and BRAMs
- Functional decomposition is applied to generate a Boolean network
- We analyzed the upper bounds for a column multiplicity:
    - An index generation function, a WS function, and a RNS

Exercise

1. (Mandatory) Show a functional decomposition of a 2-input adder
2. (Optional) Show a proof for Theorem1 shown in the lecture
3. (Optional) Show a proof for Theorem2 shown in the lecture

Send a report by a PDF file to nakahara@ict.e.titech.ac.jp

Deadline is 3rd, Aug., 2018