Random Number Generators

Selected Topics on Hardware for Security (NWI-IMC065)

Radboud University

Ileana Buhan, November 2021

What is randomness?

A

29 49 0d 57 19 0d 18 36 4c 49 54 1e 59 38 01 49 52 41 1f 3c
20 25 34 43 4c 3f 5d 44 3e 05 34 47 4a 5f 5f 46 3e 42 0e 50 3f
0a 25 64 4d 17 3a 5e 19 10 35 04 01 25 3a 0f 43 36 4f 07 25
45 30 d0 c1 82 61 b3 b3 32 d2 74 b5 52 65 42 75 82 41 82
37 39 2c 45 0b 22 16 46 08 26 5a 36 4f 42 3b 35 4e 59 51 4b
64 05 28 5c 3a 1d 19 36 43 06 33 0f 42 09 54 61 4c 2f 02 33
4b 64 02 09 0c 2b 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14 27
23 26 50 5e 45 18 07 34 60 3a 4b 5d 63 5f 3c 45 18 47 0d 14
0c 64 29 36 0c 52 01 2f 37 2e 4f 46 3f 0a 21 4f 0c 27 38 61
2a 33 08 16 17 5b 4b 4d 44 27 20 20 44 2f 5f 4b 15 4a 61 61
4d 17 3a 5e 19 10 35 04 01 25 3a 0f 43 36 4f 07 25
45 30 d0 c1 82 61 b3 b3 32 d2 74 b5 52 65 42 75 82 41 82
37 39 2c 45 0b 22 16 46 08 26 5a 36 4f 42 3b 35 4e 59 51 4b
64 05 28 5c 3a 1d 19 36 43 06 33 0f 42 09 54 61 4c 2f 02 33
4b 64 02 09 0c 2b 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14
23 26 50 5e 45 18 07 34 60 3a 4b 5d 63 5f 3c 45 18 47 00

B

59 38 01 49 52 41 1f 3c 20 25 34 43 4c 3f 5d 44 3e 05 34 47 4a 5f 5f 46
3e 42 0e 50 3f0a 25 64 4d 17 3a 5e 19 10 35 04 01 25 3a 0f 43 36 4f 07
25 45 30 d0 c1 82 61 b3 b3 32 d2 74 b5 52 65 42 75 82 41 82
37 39 2c 45 0b 22 16 46 08 26 5a 36 4f 42 3b 35 4e 59 51 4b64 05 28
5c 3a 1d 19 36 43 06 33 0f 42 09 54 61 4c 2f 02 33 4b 64 02 09 0c 2b
5b 48 50 20 18 22 30 35 4f 5e 06 2f 14 27 23 26 50 5e 45 18 07 34 60
3a 4b 5d 63 5f 3c 45 18 47 0d 14 0c 64 29 36 0c 52 01 2f 37 2e 4f 46
3f 0a 21 4f 0c 27 38 61 2a 33 08 16 17 5b 4b 4d 44 27 20 20 44 2f 5f
4b 15 4a 61 61 4d 17 3a 5e 19 10 35 04 01 25 3a 0f 43 36 4f 07 25
45 30 d0 c1 82 61 b3 b3 32 d2 74 b5 52 65 42 75 82 41 82 37 39 2c 45
0b 22 16 46 08 26 5a 36 4f 42 3b 35 4e 59 51 4b 64 05 28 5c 3a 1d 19
36 43 06 33 0f 42 09 54 61 4c 2f 02 33 4b 64 02 09 0c 2b 5b 48 50 20
18 22 30 35 4f 5e 06 2f 14 23 26 50 5e 45 18 07 34 60 3a 4b 5d 63 5f
3c 45 18 47 73 a5 ed 19 10 35 04 34 e5 09 a3 25 17 05 a1 f0 37 11
47 5d 07 3c02 0f 21 3b 14 20 2d 15 1c 5d 44 10 2a 23 09 30 0c 13
25 43 28 3530 19 44 5a 55 32 59 32 4e 49 1d 37 1c 1d 0f 2b 35 44 5b

C

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14 23 26 50 5e 45 18 07 34
3a 4b 5d 63 5f 3c 45 18 47 73 a5 ed 19 10 35 04 34 e5 09 a3 25 17
a1 f0 37 11 aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa
B0 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14 23 26 50 5e 45 18 07 34
3a 4b 5d 63 5f 3c 45 18 47 73 a5 ed 19 10 35 04 34 e5 09 a3 25 17
a1 f0 37 11 aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
B0 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14 23 26 50 5e 45 18 07 34
3a 4b 5d 63 5f 3c 45 18 47 73 a5 ed 19 10 35 04 34 e5 09 a3 25 17
a1 f0 37 11 aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa aa
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

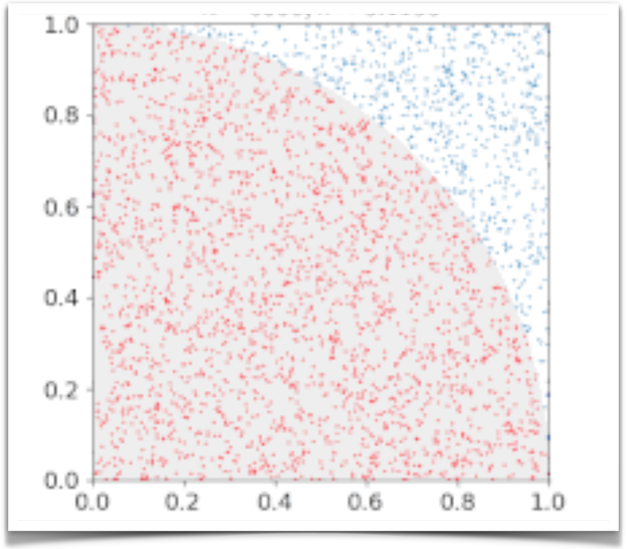
Overview

1. Why is randomness important?
2. What is randomness?
3. How are random numbers generated?
4. Entropy sources
5. Entropy extractors
6. How do we evaluate random number generators?

Why is randomness important?

Why is randomness important?

Simulation



Source: wikipedia: Monte Carlo estimation
to compute the value of Π

Computer grafics

Source: https://thebookofshaders.com/13/
use randomness to
create realistic landscapes

Random sampling

Source: https://www.displayr.com/
what-is-random-sampling/

Who is the author?

“Can’t rely on the secrecy of the algorithms,
only of the secret keys”

AUTHOR?

Who is the author?

“Can’t rely on the secrecy of the algorithms,
only of the secret keys”

AUGUST KERCKHOFF

Why is randomness important for security?

Why is randomness important for security?

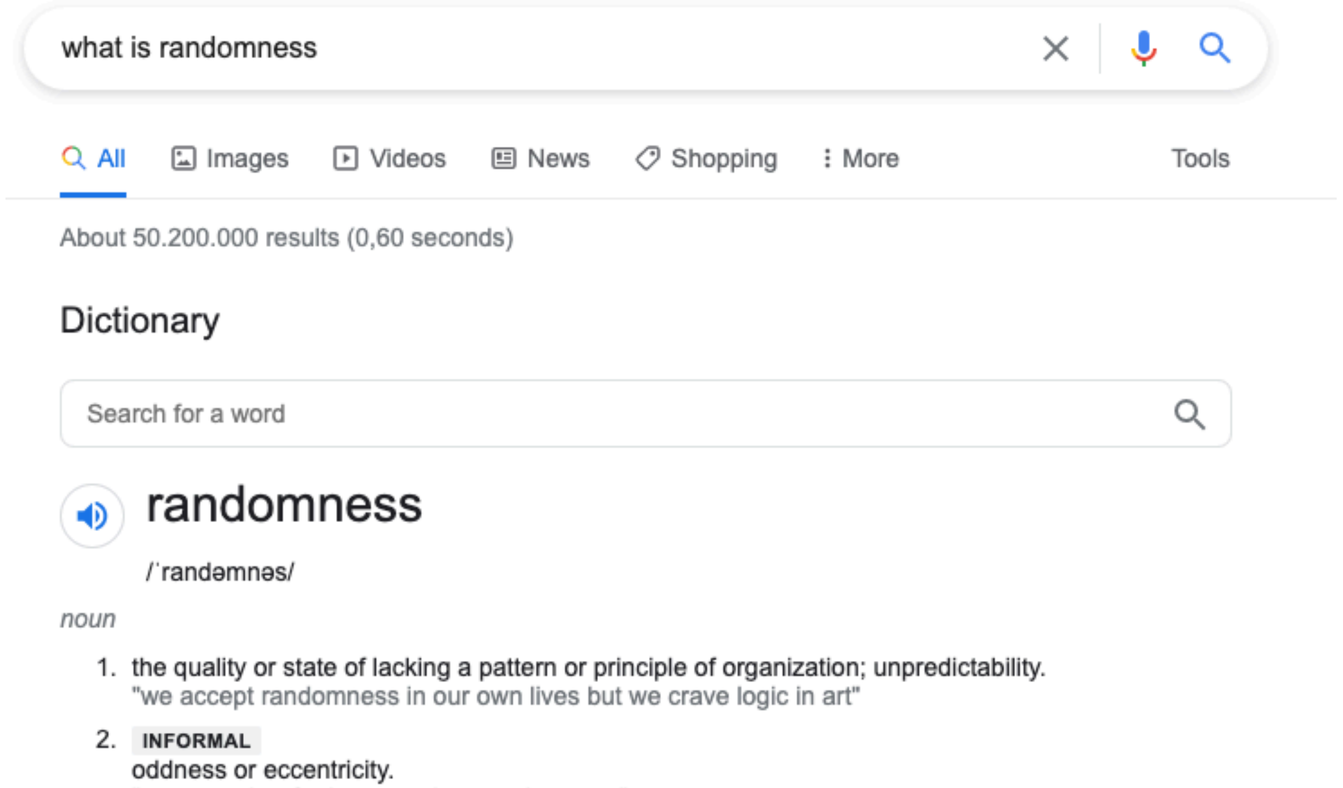
1. Key generation
2. Countermeasures (masking, hiding, code obfuscation)
3. Unpredictability (random challenges)
4. Freshness (nonce, non-repeat)
5. Noise
6. …

Take away

Without randomness **cryptography is impossible** because all operations would be predictable and therefore insecure.

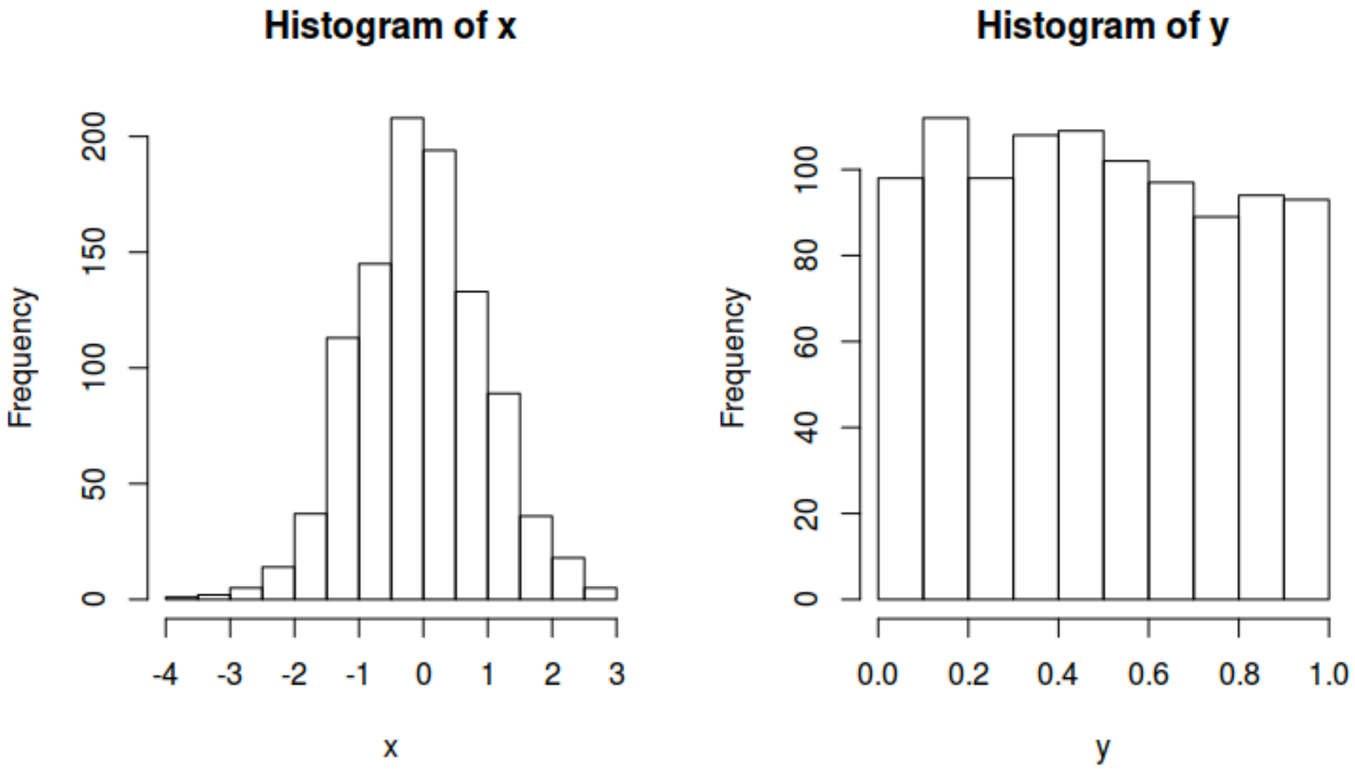
What is randomness?

What is randomness?



- Random bits
- Random process
- Probability distribution

Randomness as a Probability Distribution



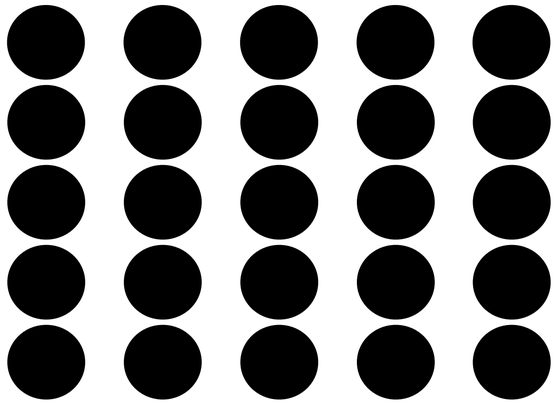
What is randomness?

Entropy - a measure of randomness

Entropy is a scientific concept, as well as a measurable physical property that is most commonly associated with a state of disorder, randomness, or uncertainty.

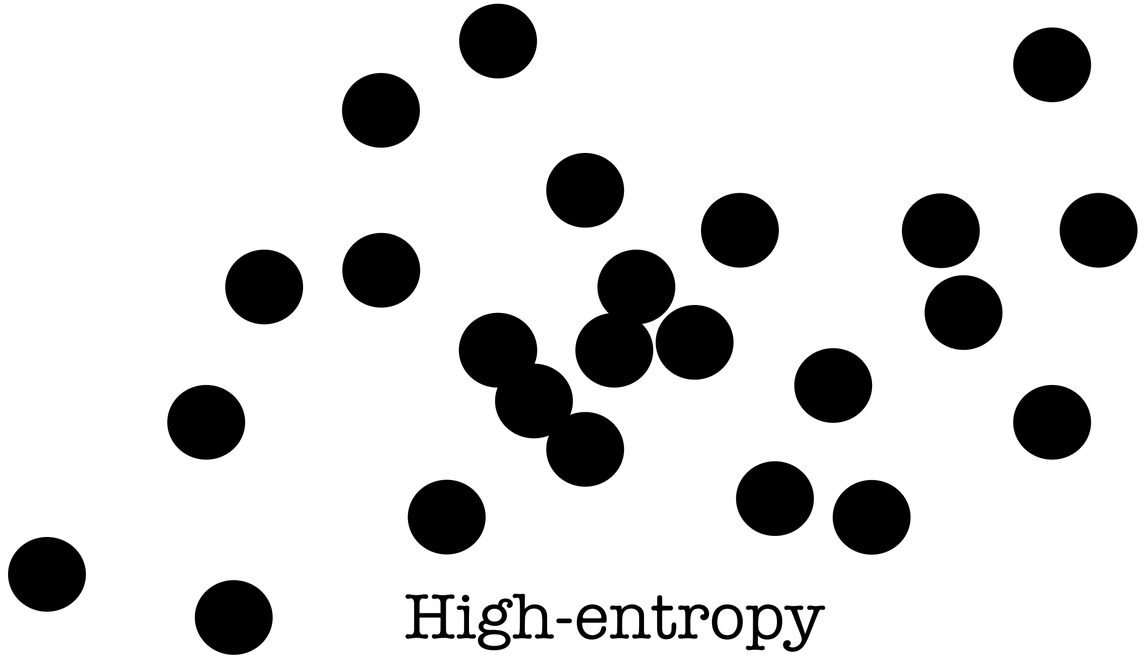
Wikipedia

Entropy



Low-entropy

Entropy

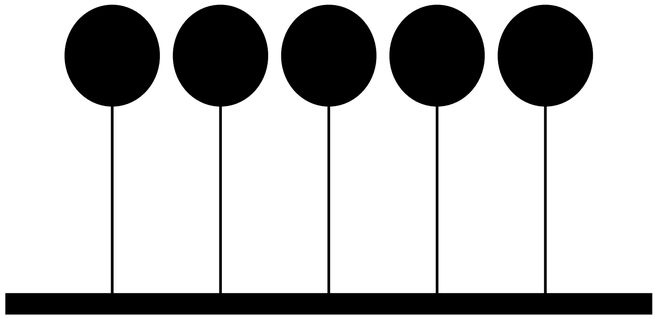


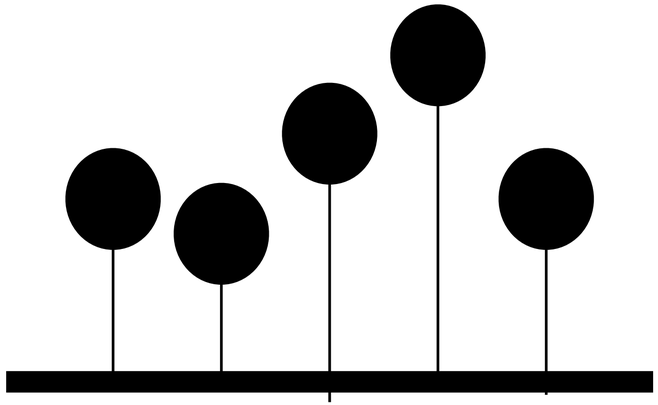
Events

Independent events

Dependent events

Indentically distributed





Take away

Good randomness is IID extract from a high entropy source.

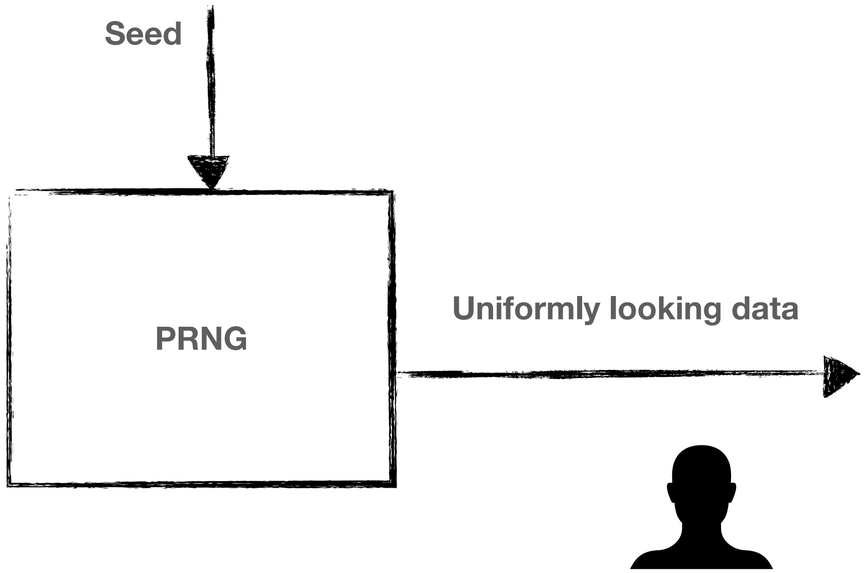
How are random numbers generated?

Random number generators

TRNG	True Random Number Generator	Physical source of entropy	Expensive, Slow
PRNG	Pseudo-Random Number Generator	Deterministic function	Fastest
CS-PRNG	Cryptographically Secure Pseudo Random Number Generator	Deterministic function	Fast, Backtracking resistance, Forward prediction resistance

How are random numbers generated?

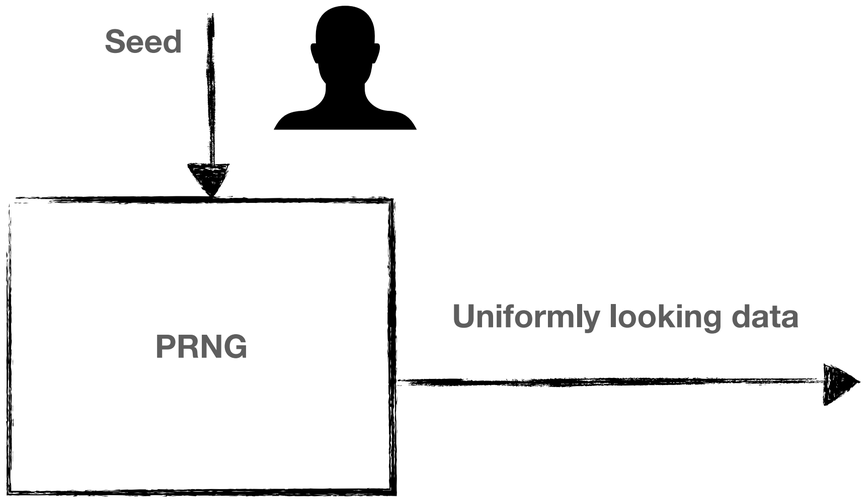
Unpredictability



Unpredictabilty is a property of what the observer knowns.

How do we evaluate RNGs?

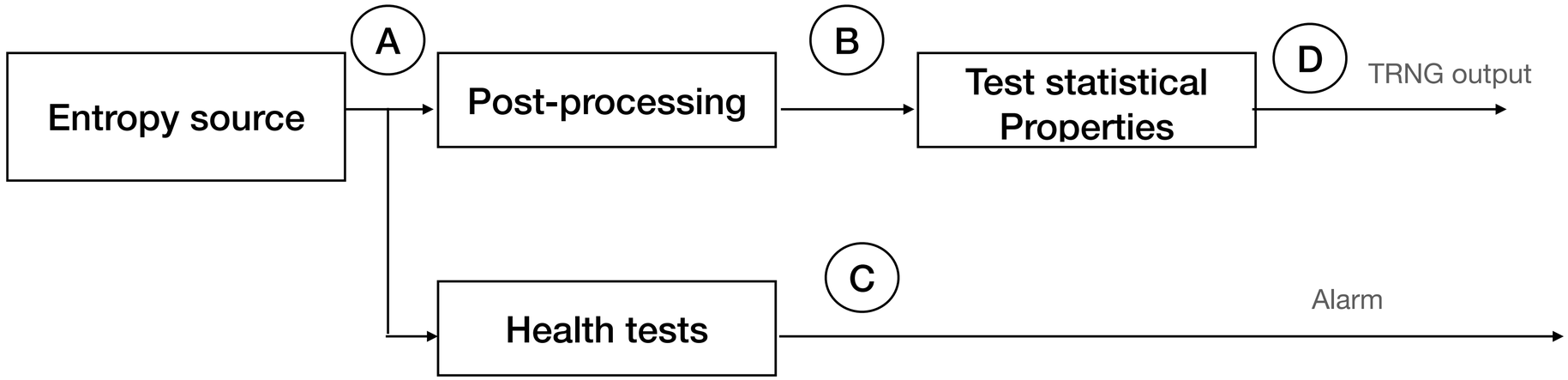
Unpredictability



Unpredictabilty is a property of what the observer knowns.

How do we evaluate RNGs?

Anatomy of a secure TRNG implementation



A — Raw data, unprocessed random values

B — Whitened/Cooked data

D — Quality of the whitened data

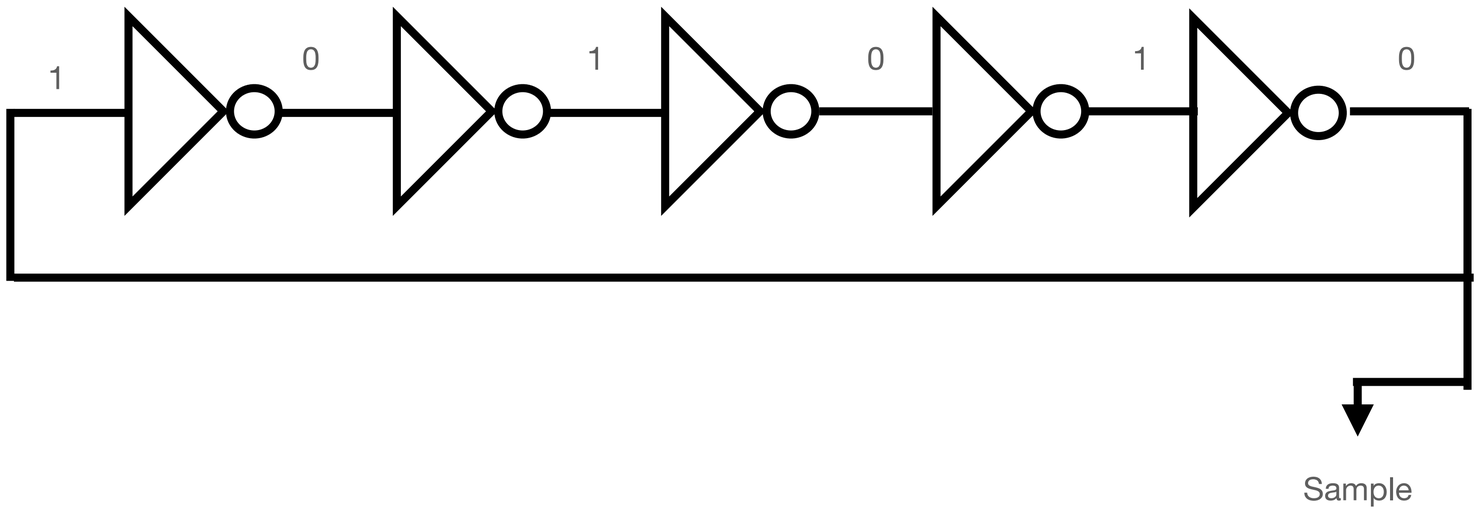
C — Health tests of the raw data

Take away

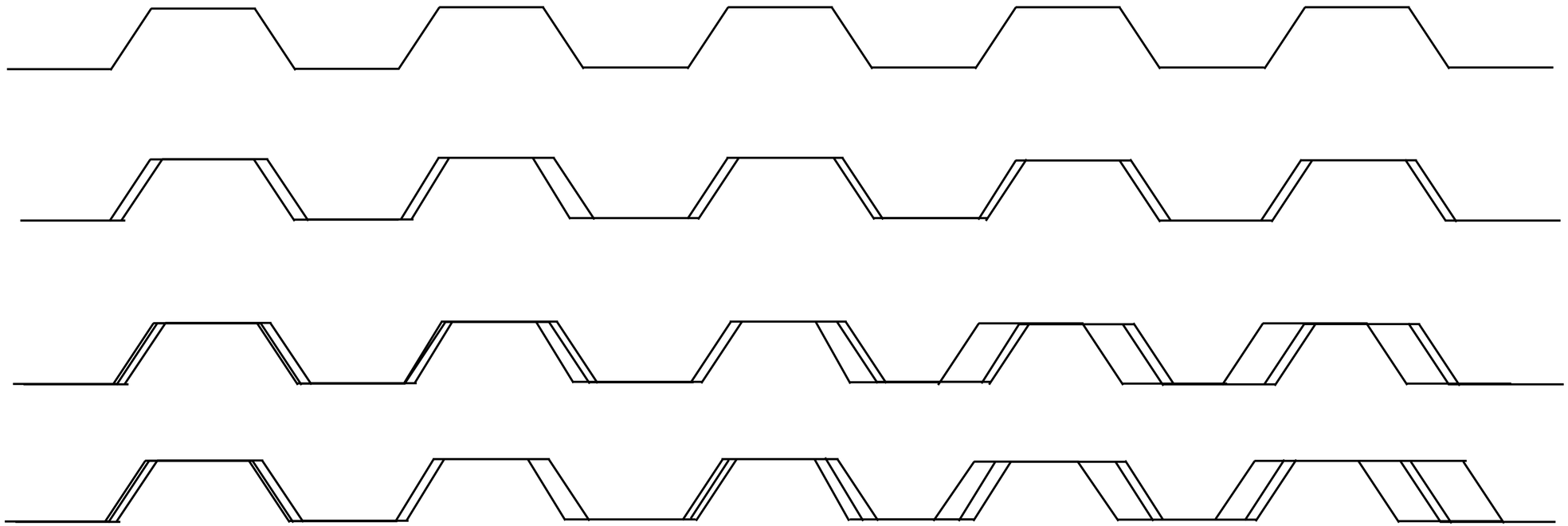
A random number generator suitable for cryptographic use is **unpredictable.**

Entropy sources

Ring Oscillator (RO) - entropy sources



Ring Oscillator (RO) - entropy sources



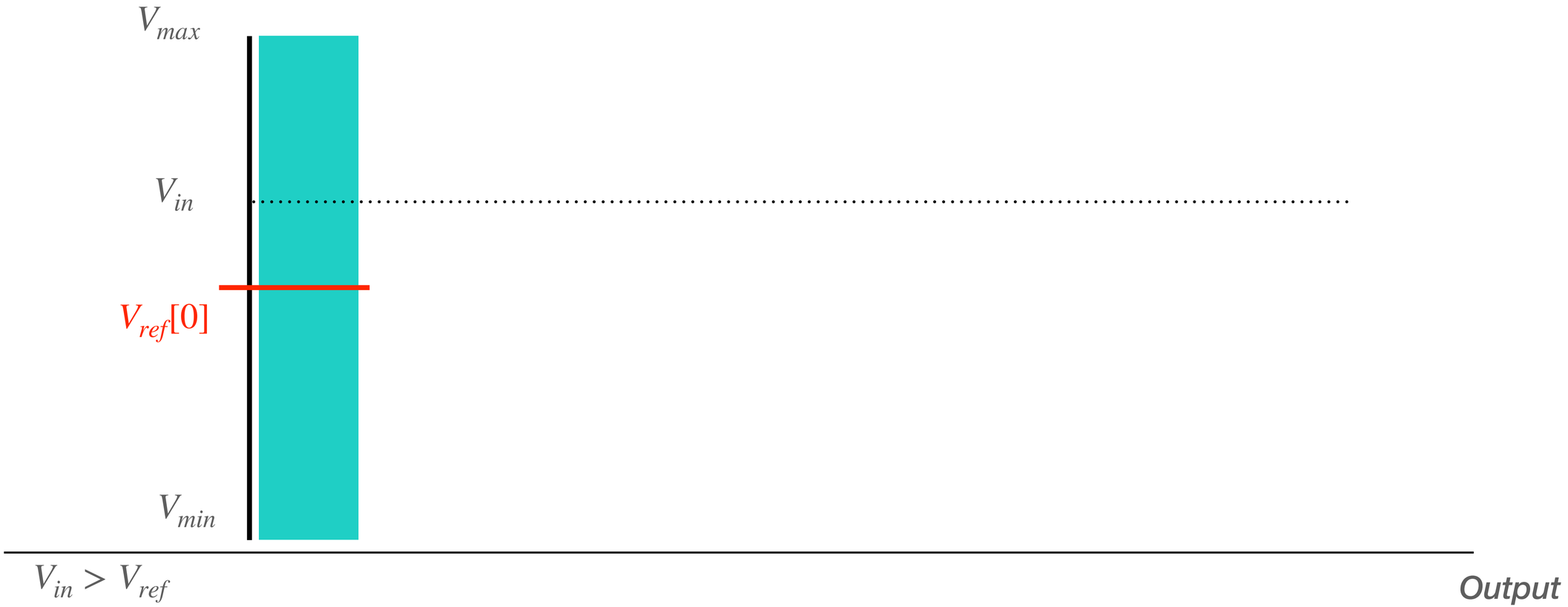
Modular Noise Multiplier/ADC

https://www.crowdsupply.com/13-37/infinite-noise-trng

Modular Noise Multiplier/ADC

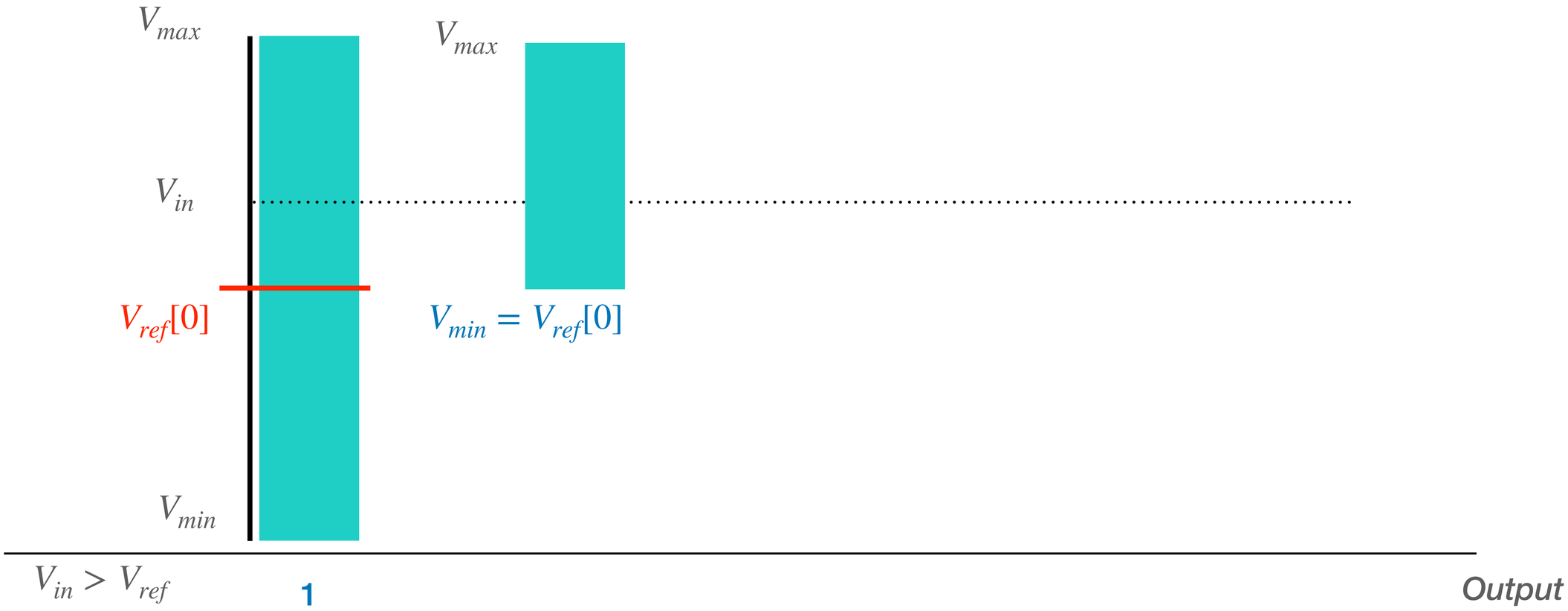
Modular Noise Multiplier/ADC

Modular Noise Multiplier/ADC

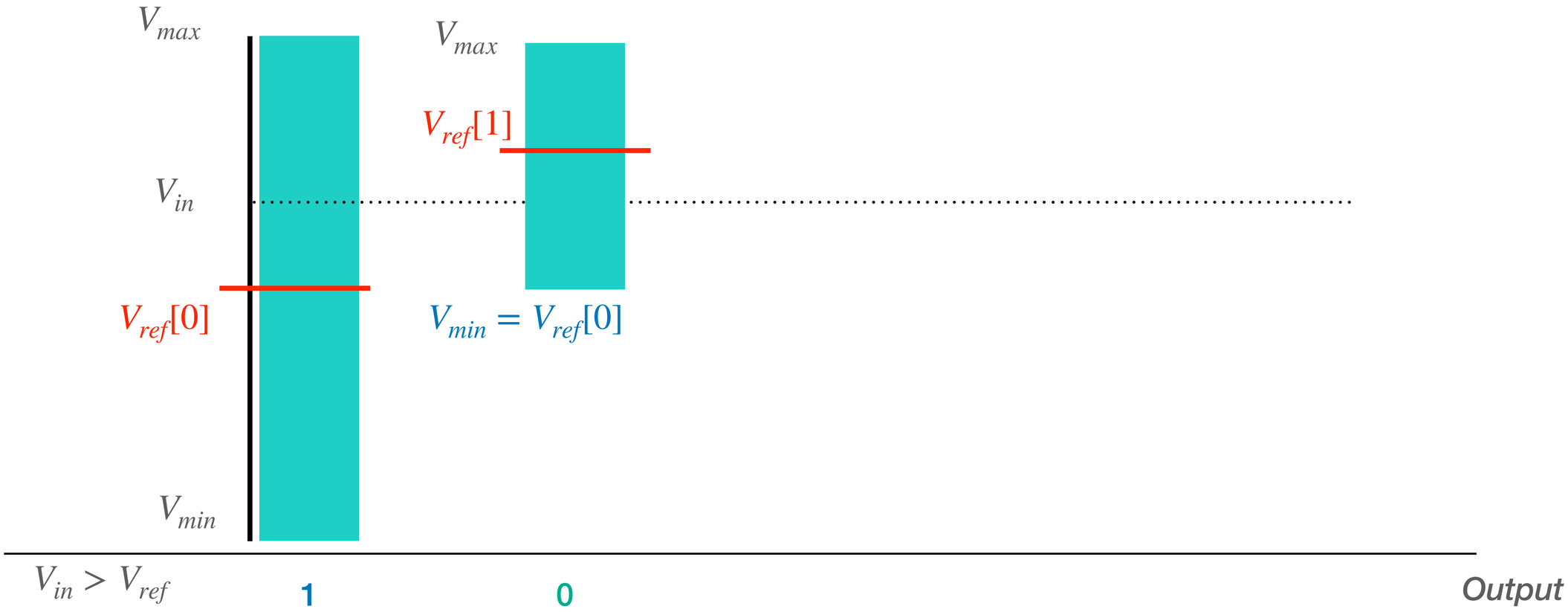


Modular Noise Multiplier/ADC

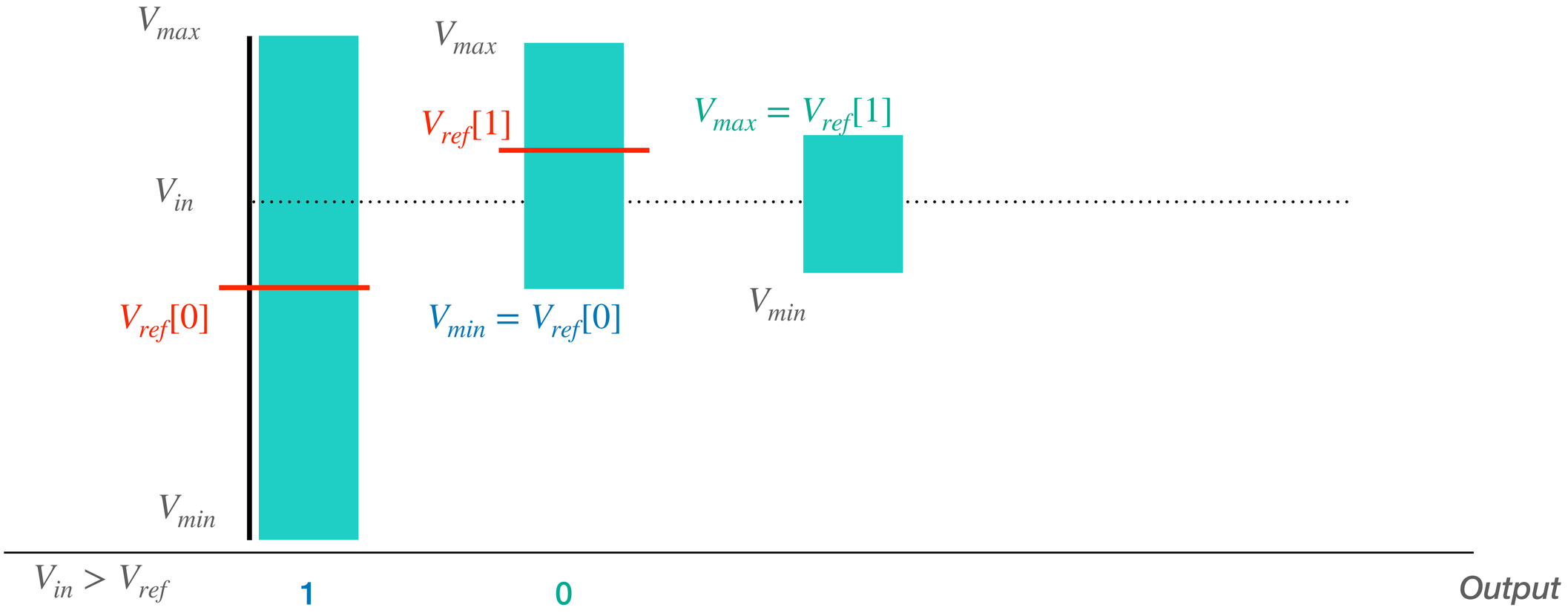
Modular Noise Multiplier/ADC



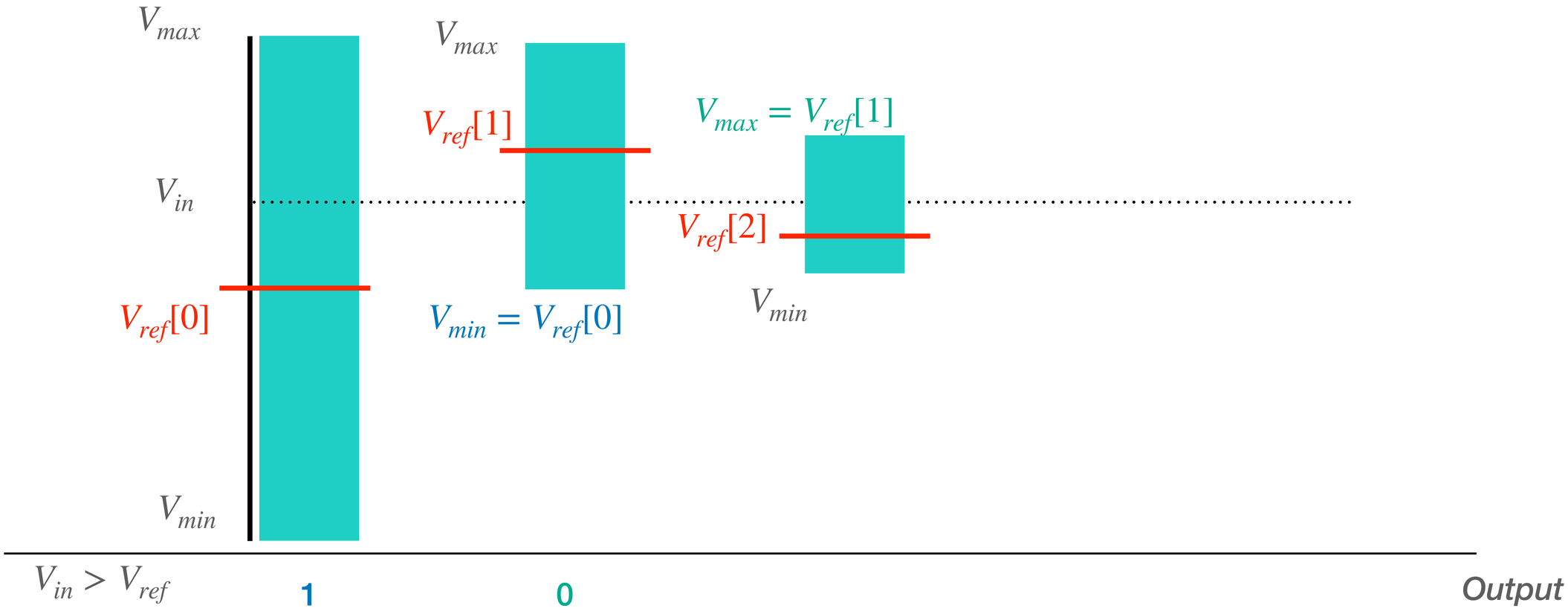
Modular Noise Multiplier/ADC



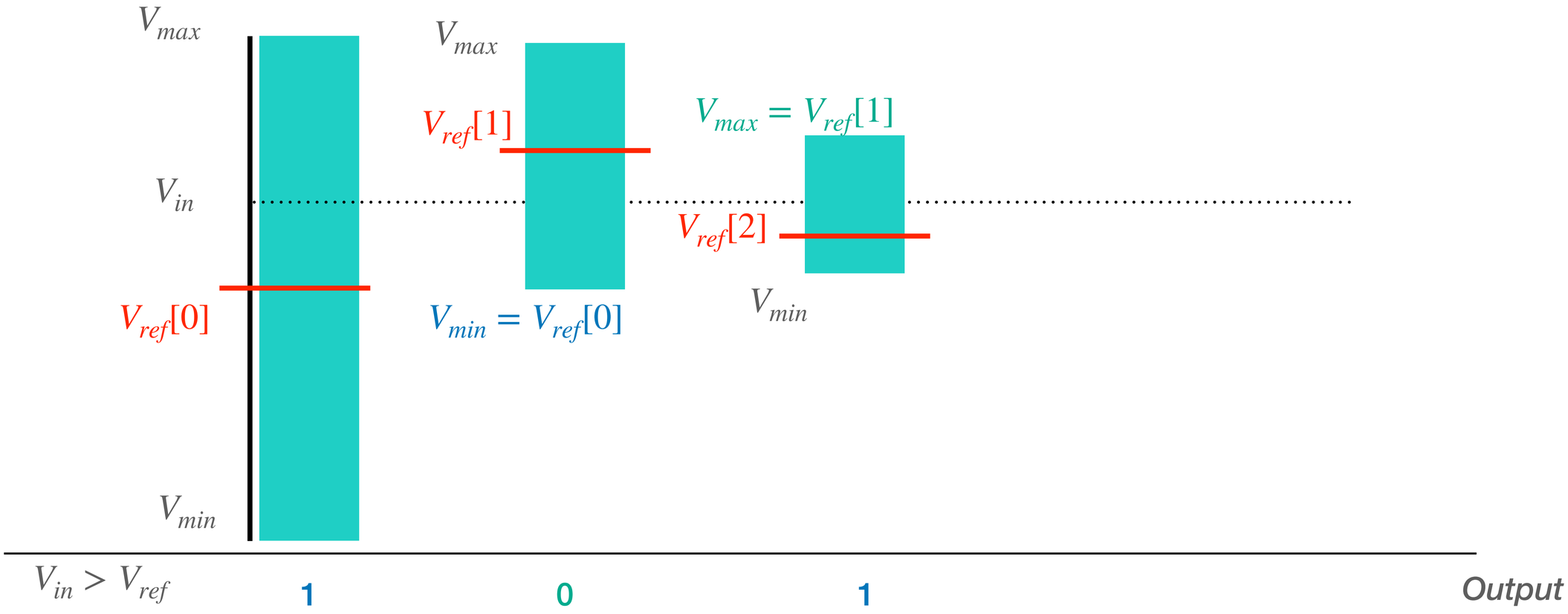
Modular Noise Multiplier/ADC



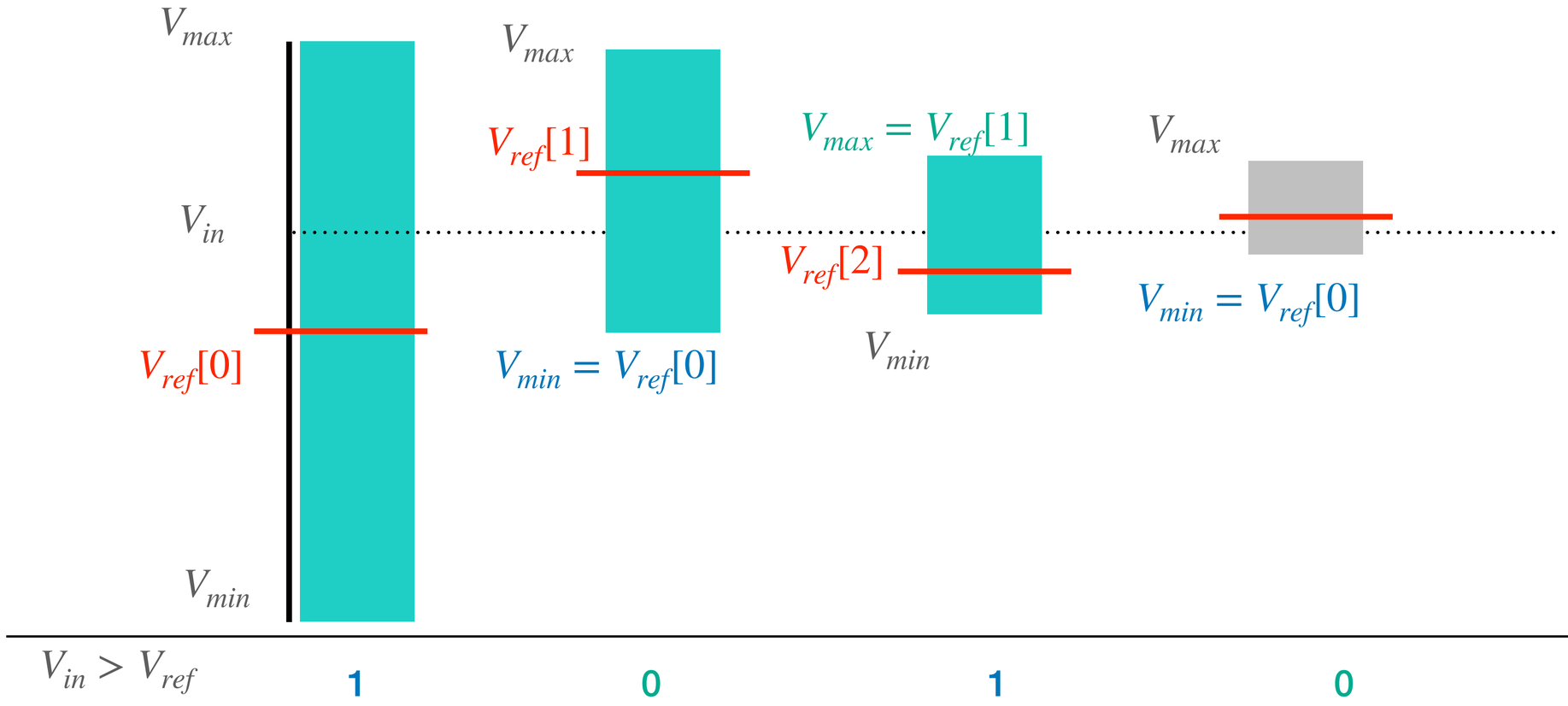
Modular Noise Multiplier/ADC



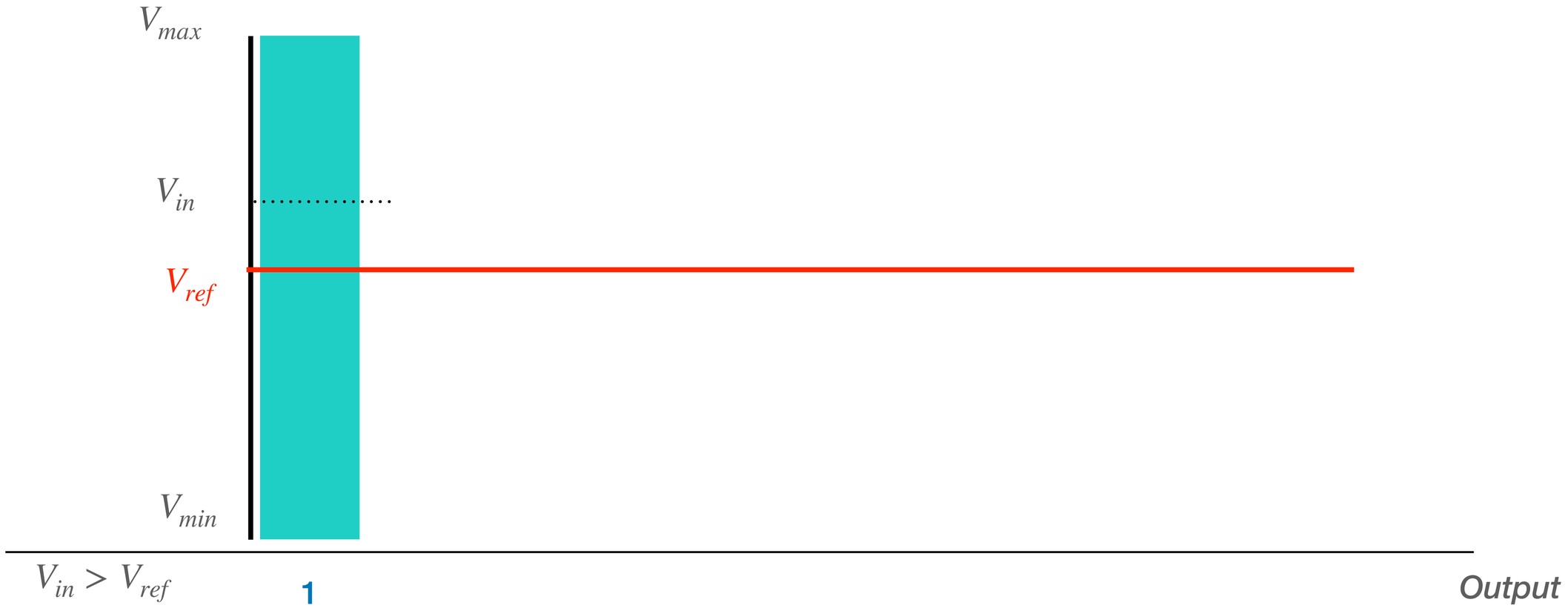
Modular Noise Multiplier/ADC



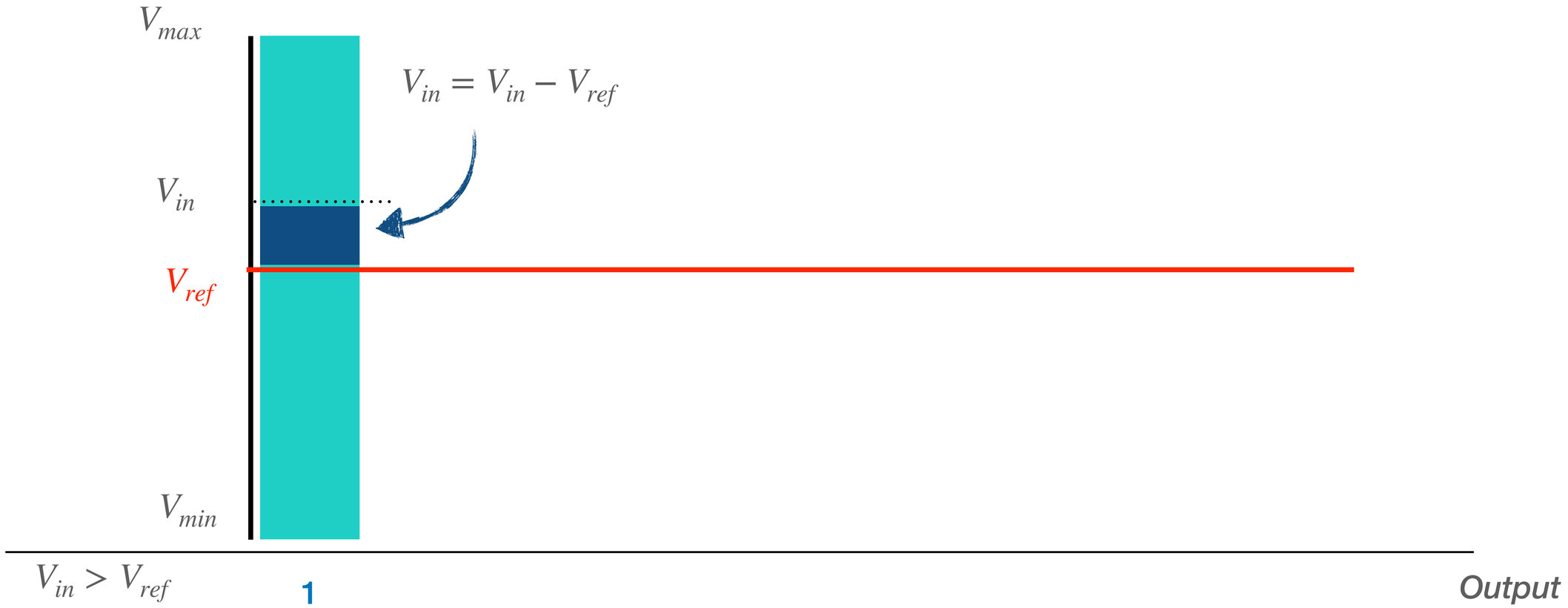
Modular Noise Multiplier/ADC



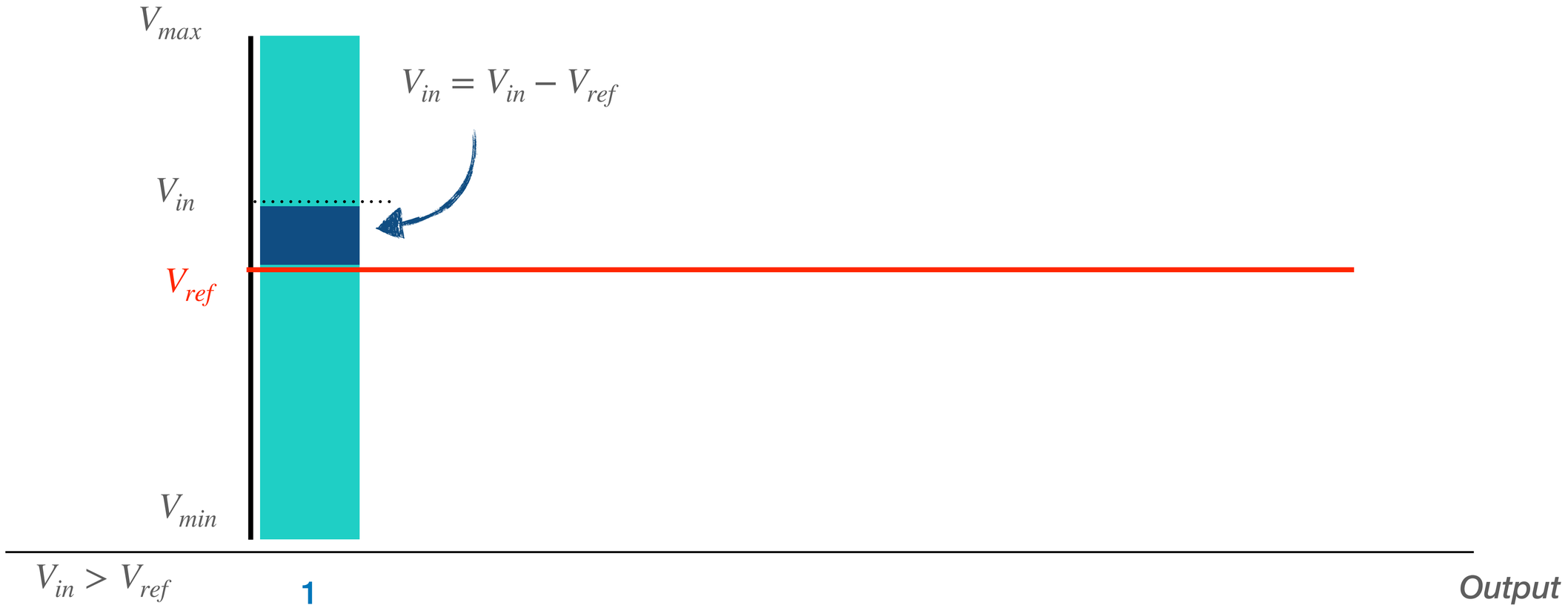
Modular Noise Multiplier



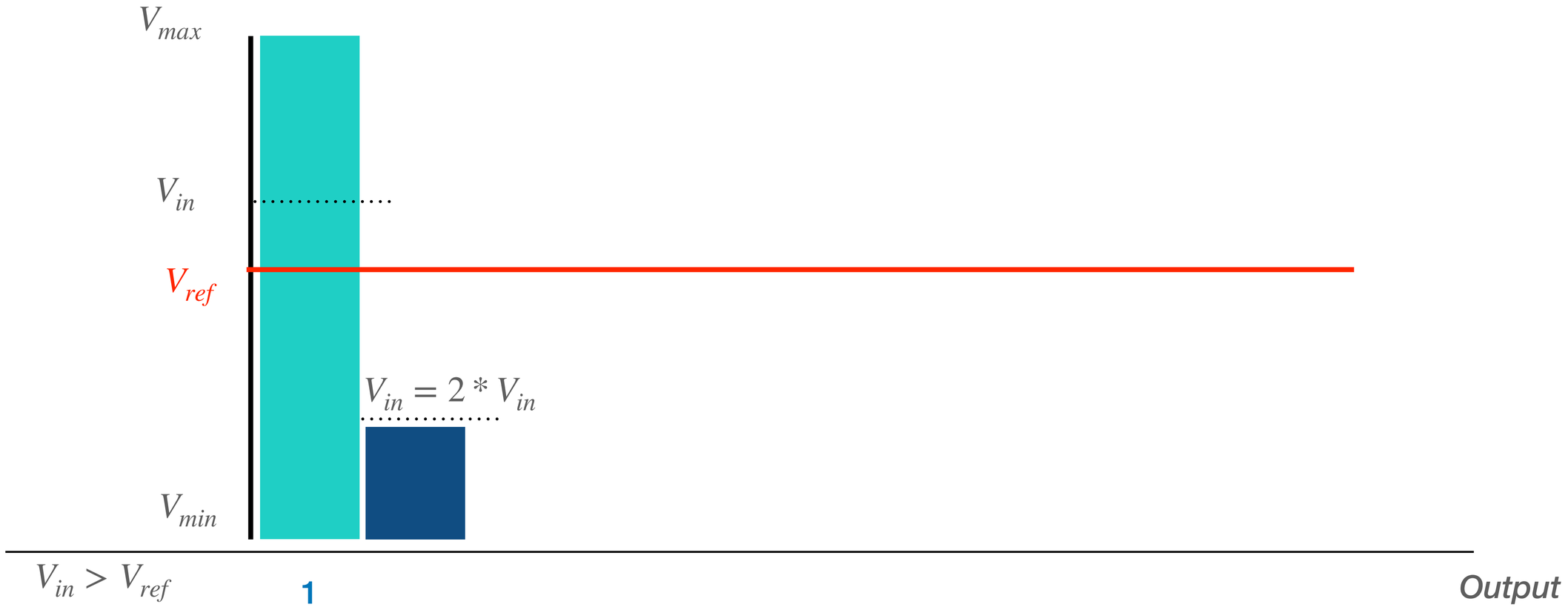
Modular Noise Multiplier



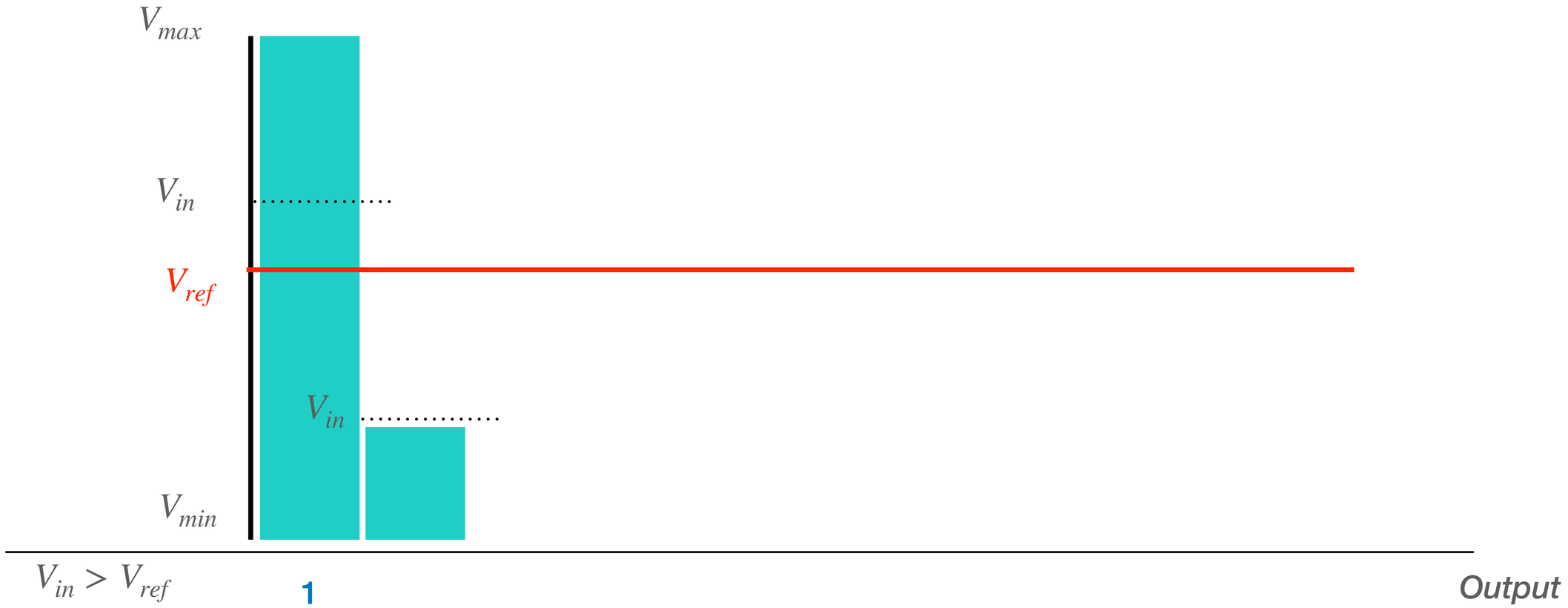
Modular Noise Multiplier



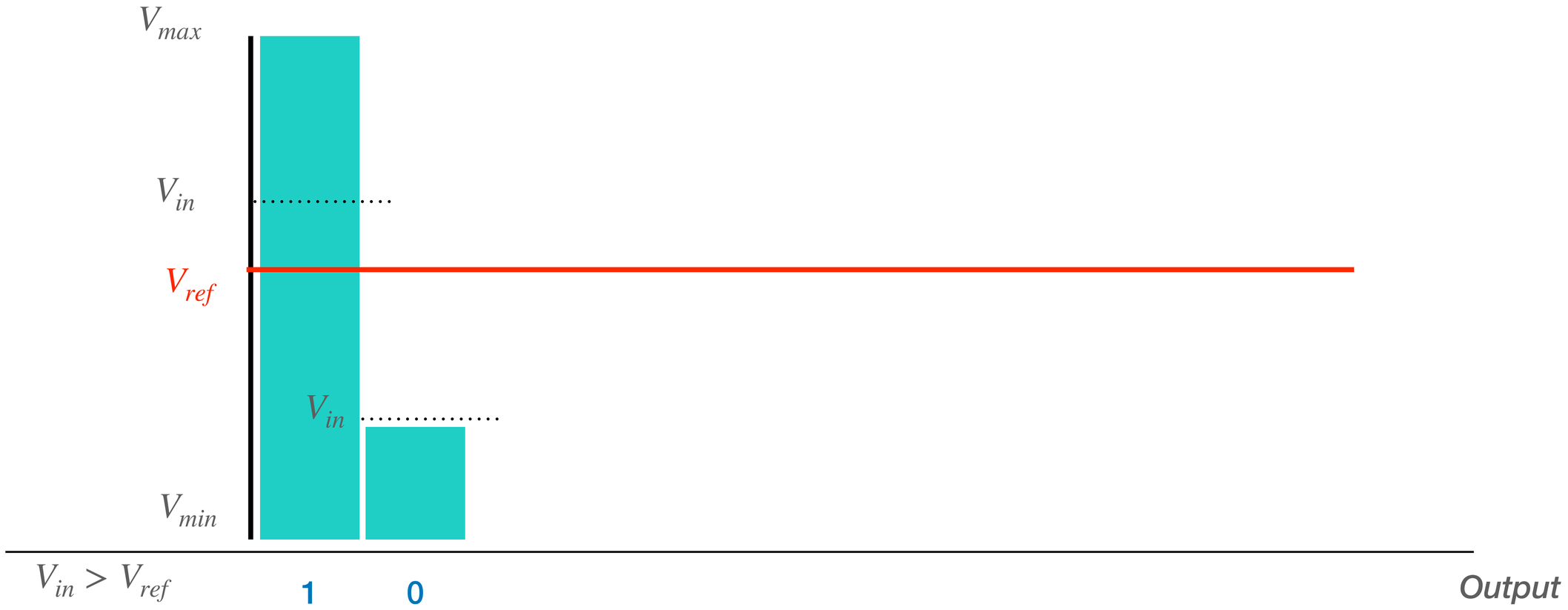
Modular Noise Multiplier



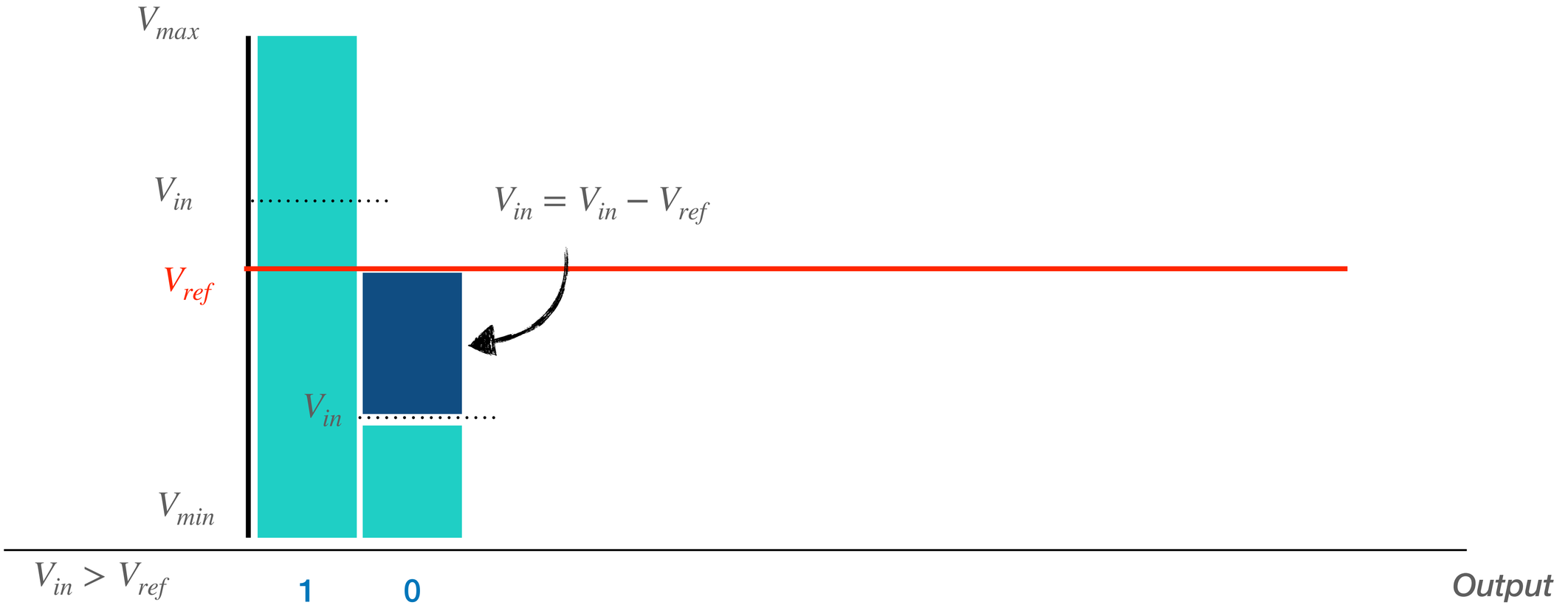
Modular Noise Multiplier



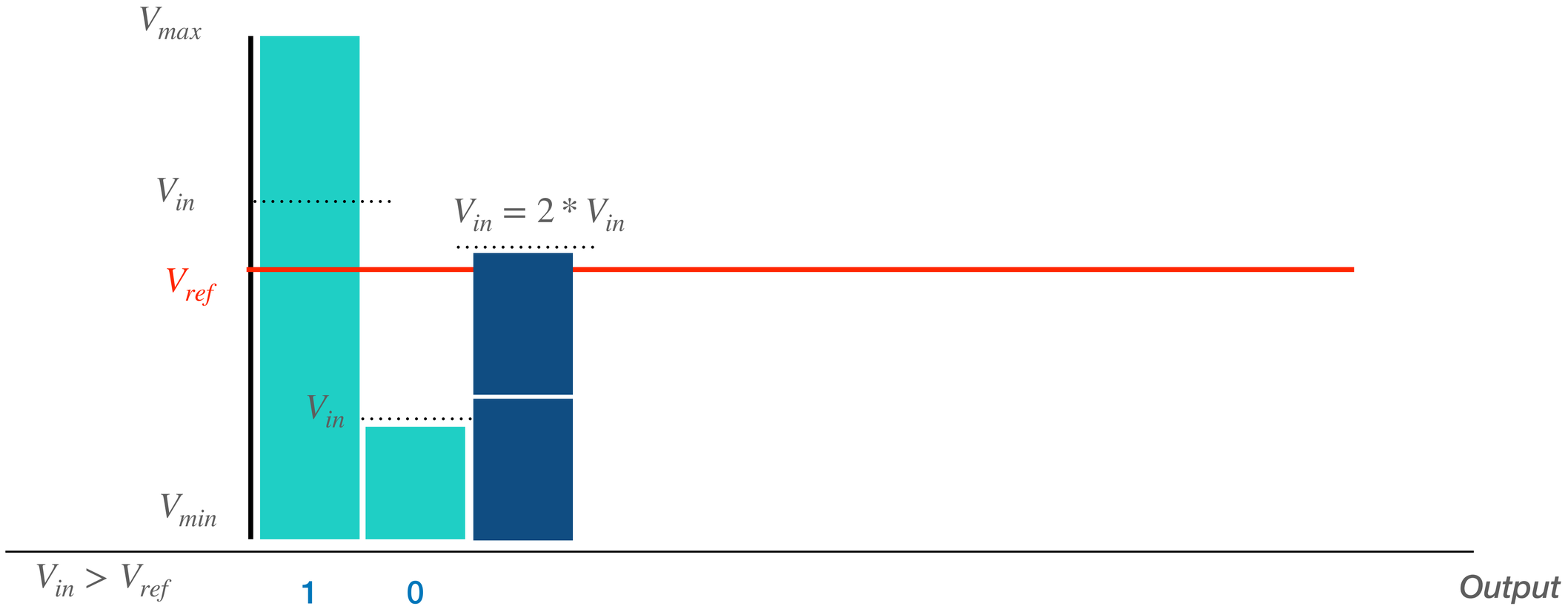
Modular Noise Multiplier



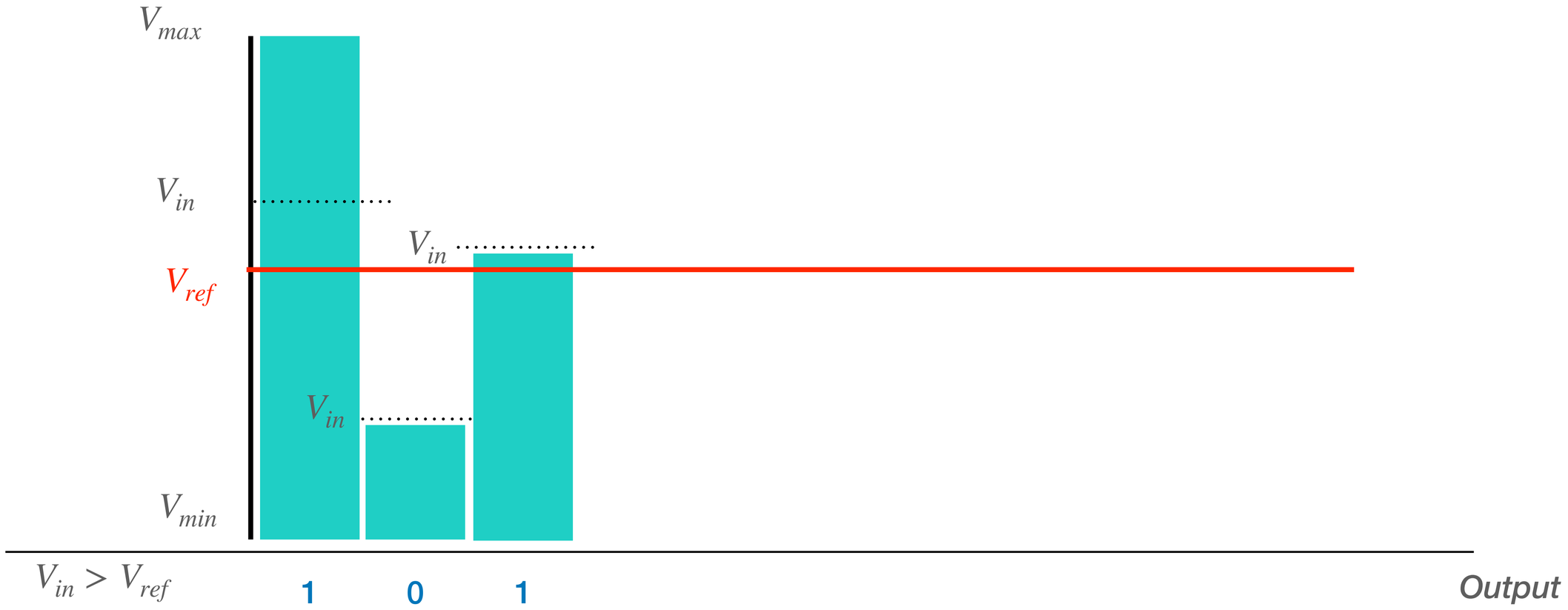
Modular Noise Multiplier



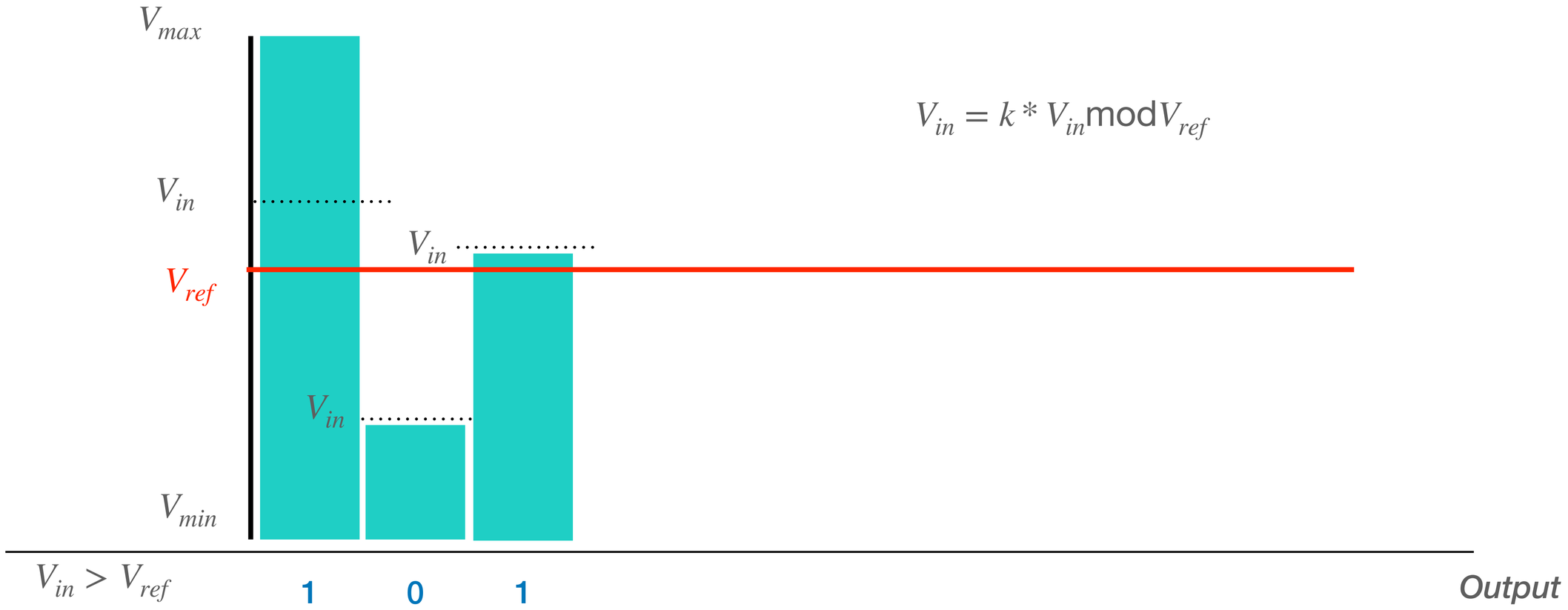
Modular Noise Multiplier



Modular Noise Multiplier



Modular Noise Multiplier



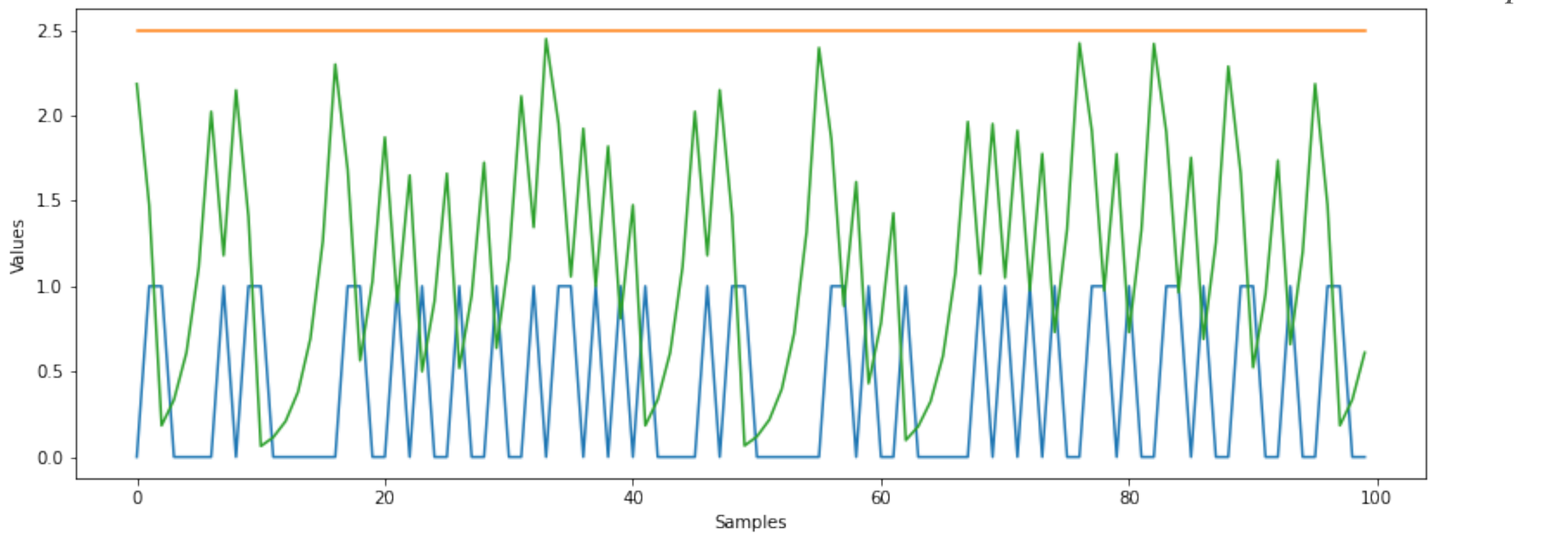
$$V_{in} = k * V_{in} \text{mod} V_{ref}$$

Modular Noise Multiplier

$V_{in} = k * V_{in} \text{mod} V_{ref}$

$k = 1{,}82$

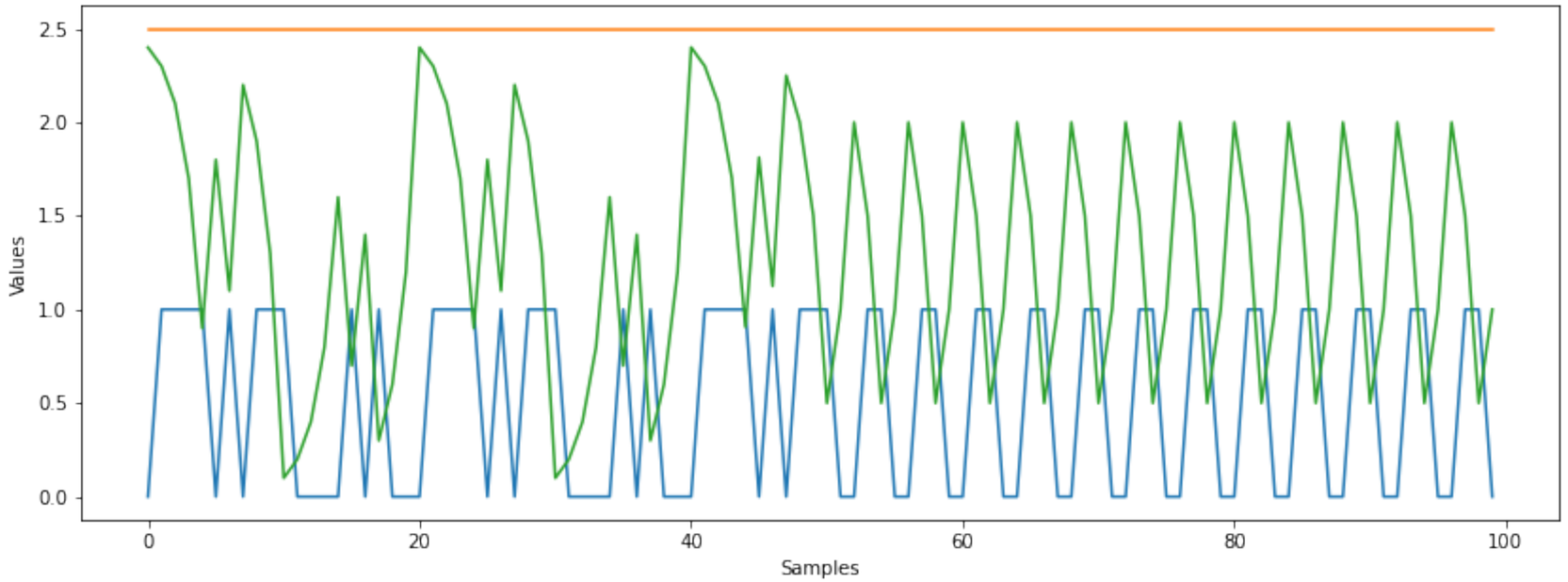
$V_{in} = 3{,}7$



Modular Noise Multiplier

$$V_{in} = k * V_{in} \text{mod} V_{ref}$$
$$k = 2$$
$$V_{in} = 3{,}7$$

V_{ref}
V_{in}
$output$



Post-processing

Is it possible to obtain a fair toss from an unfair coin?

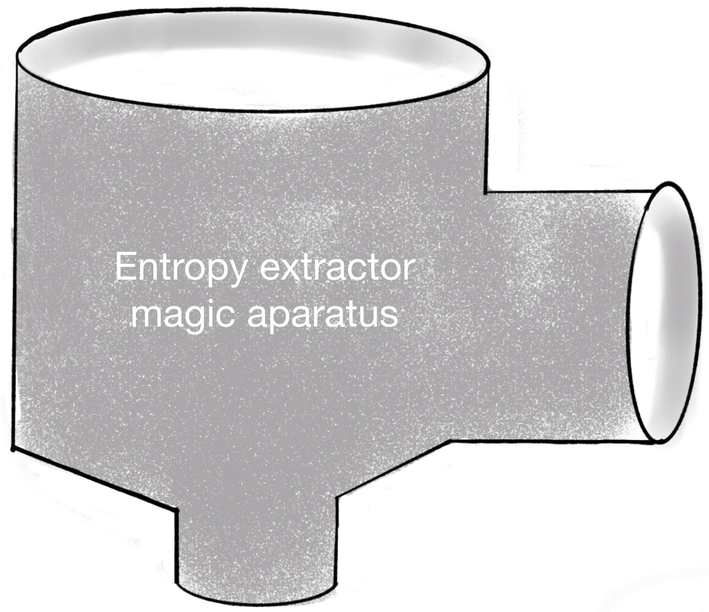
Post-processing

Is it possible to obtain a fair toss from an unfair coin?

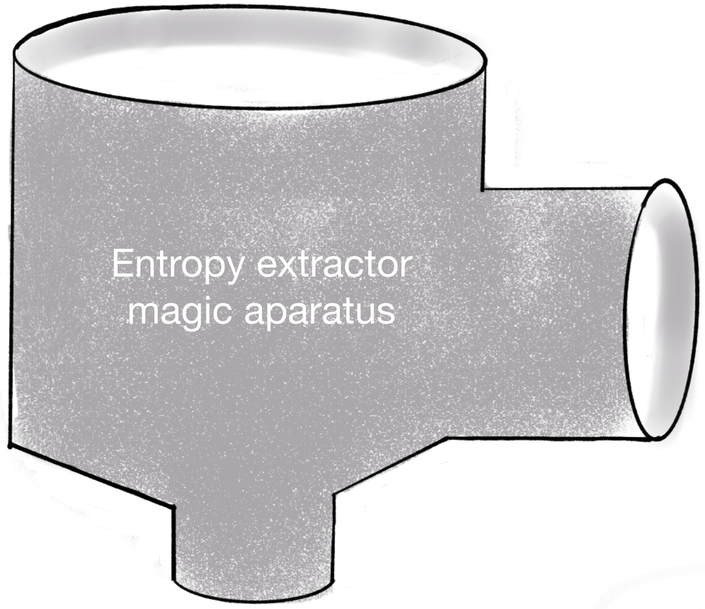
Yes, entropy extractors

Entropy Extractors

1 0 0 1 1 1 1 1 0 0



Entropy Extractors



1 0 1 1 0 1

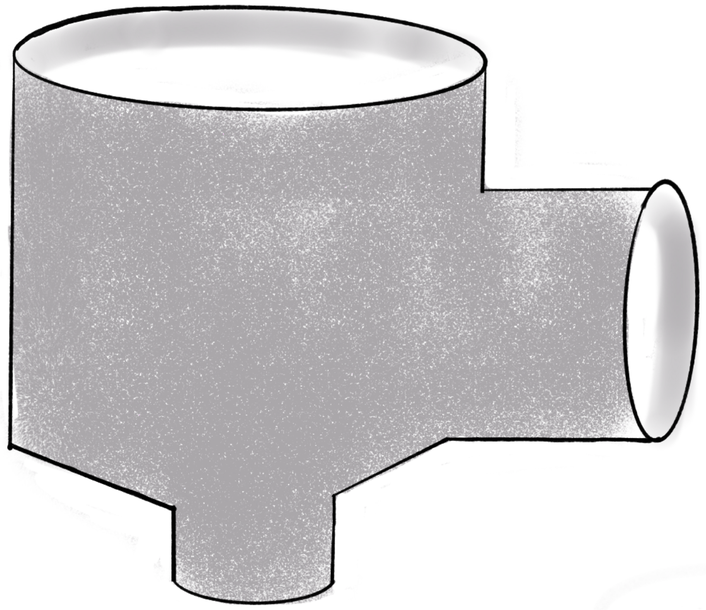
1 0 1 0

Entropy extractors

Entropy rate = $\frac{entropy}{\#bits}$

1 0 0 1 1 1 1 1 0 0

Imperfect random bits



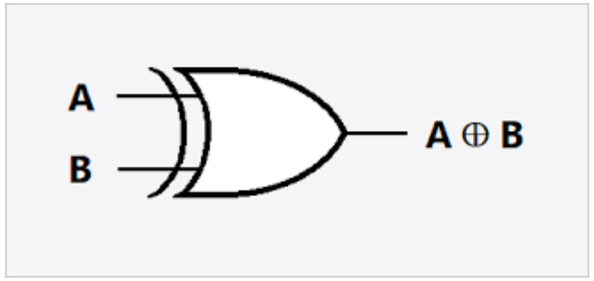
Uniformly distributed bits

1 0 1 1 0 1

1 0 1 0

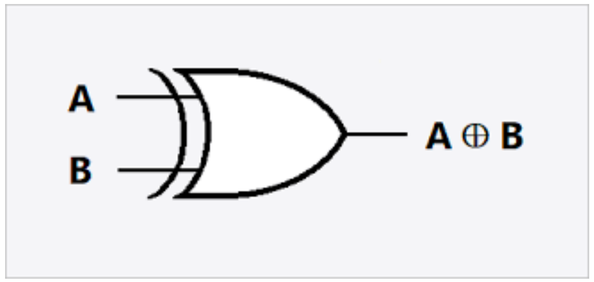
Waist, redundant bits

Entropy Extractors - a simple extractor

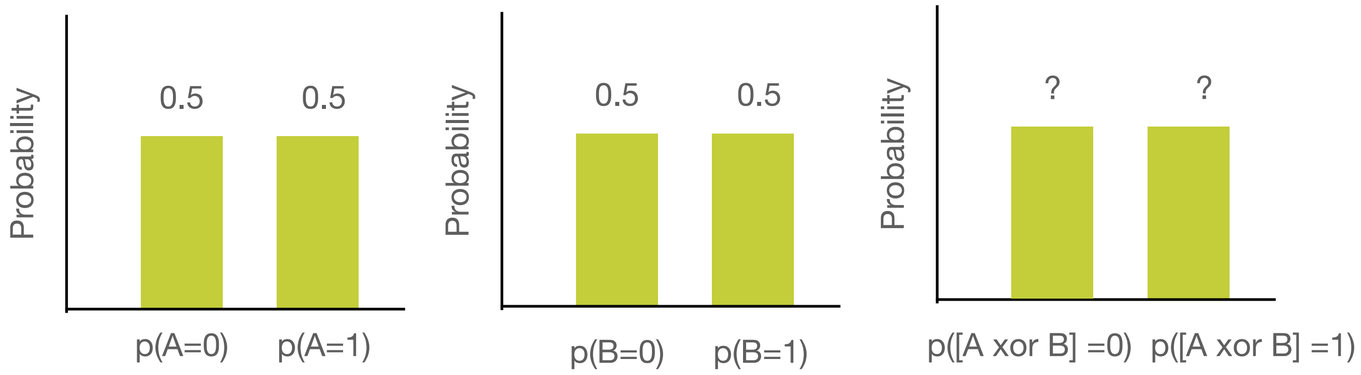


A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0

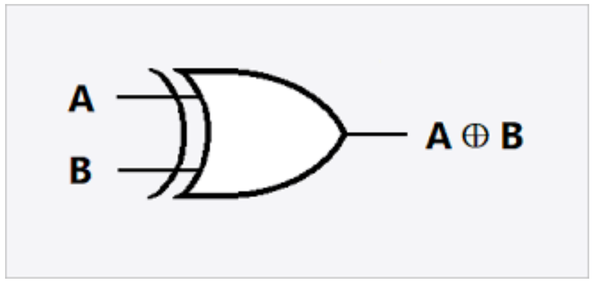
Entropy Extractors - a simple extractor

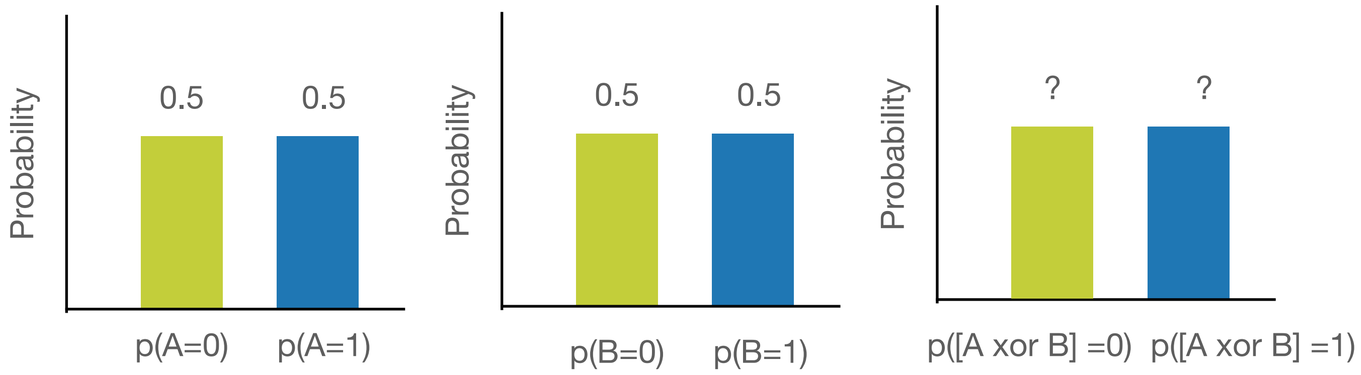


A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0



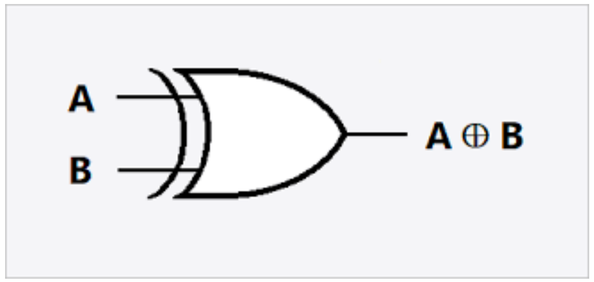
Entropy Extractors - a simple extractor

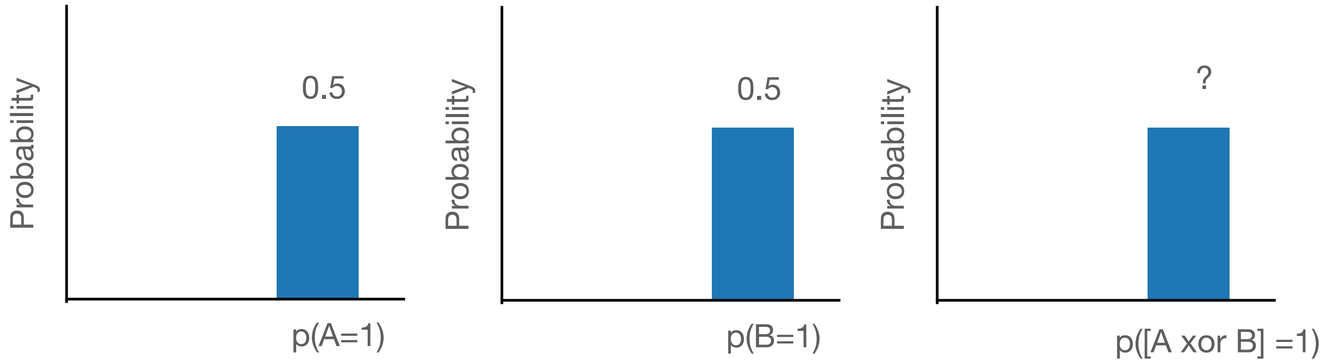




A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0

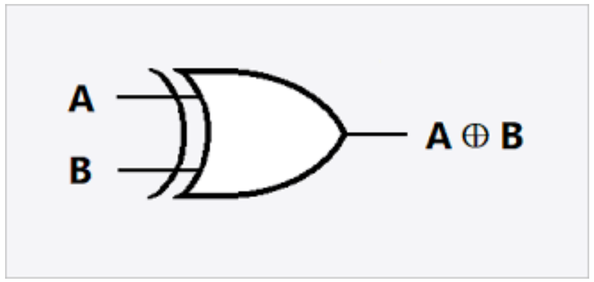
Entropy Extractors - a simple extractor



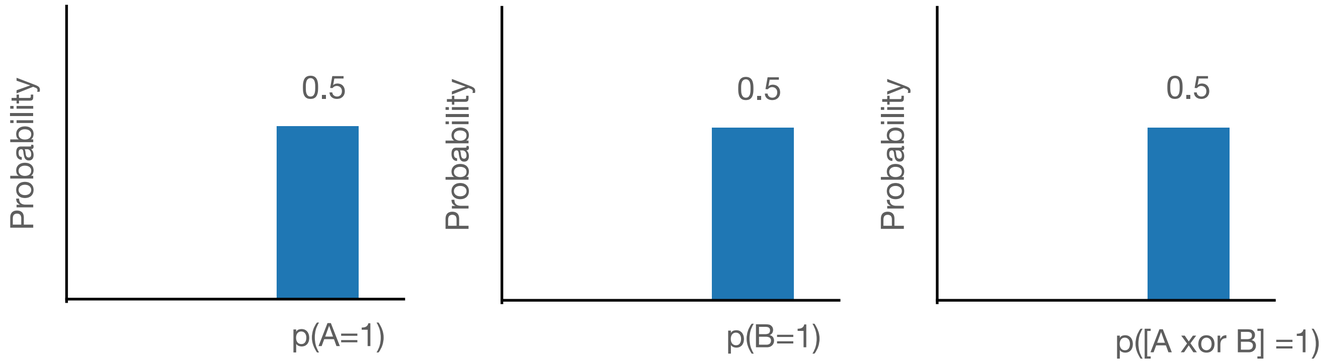


A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0

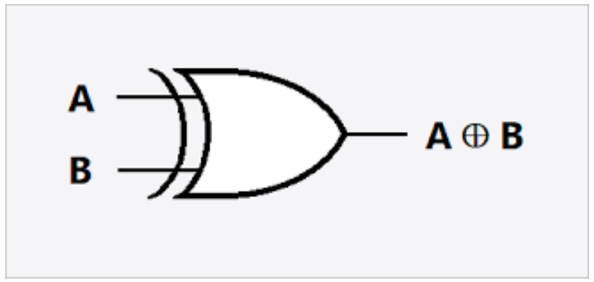
Entropy Extractors - a simple extractor

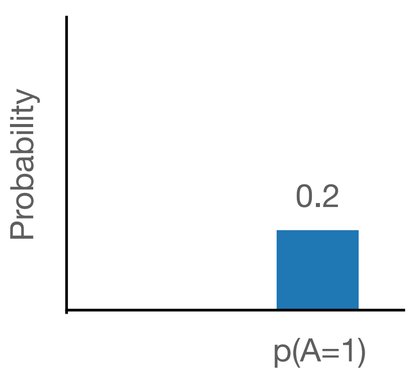


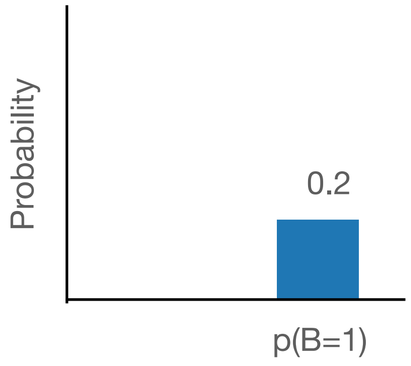
A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0

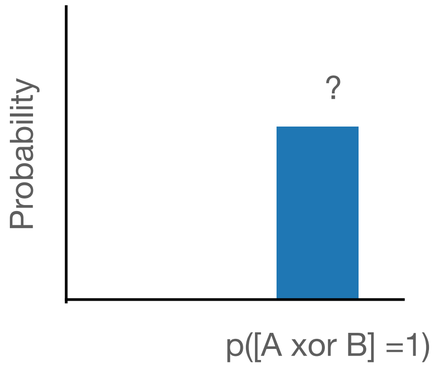


Entropy Extractors - a simple extractor



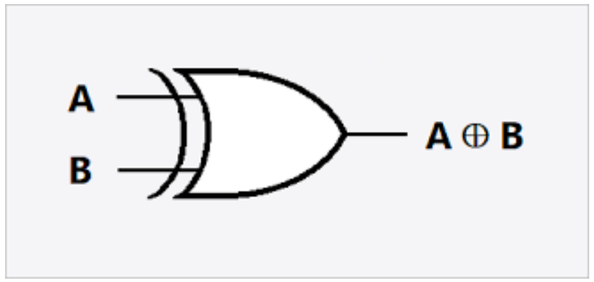




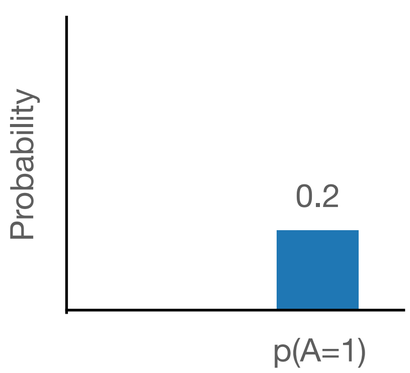


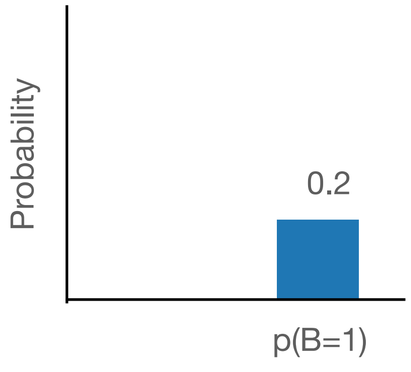
A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0

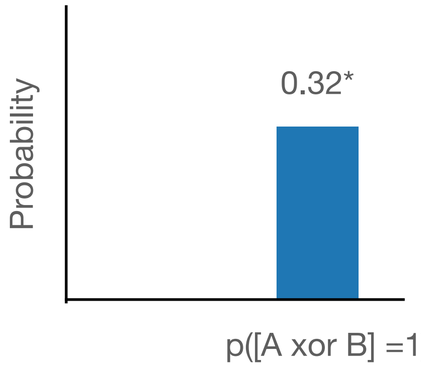
Entropy Extractors - a simple extractor



A	B	A xor B
0	0	0
1	0	1
0	1	1
1	1	0

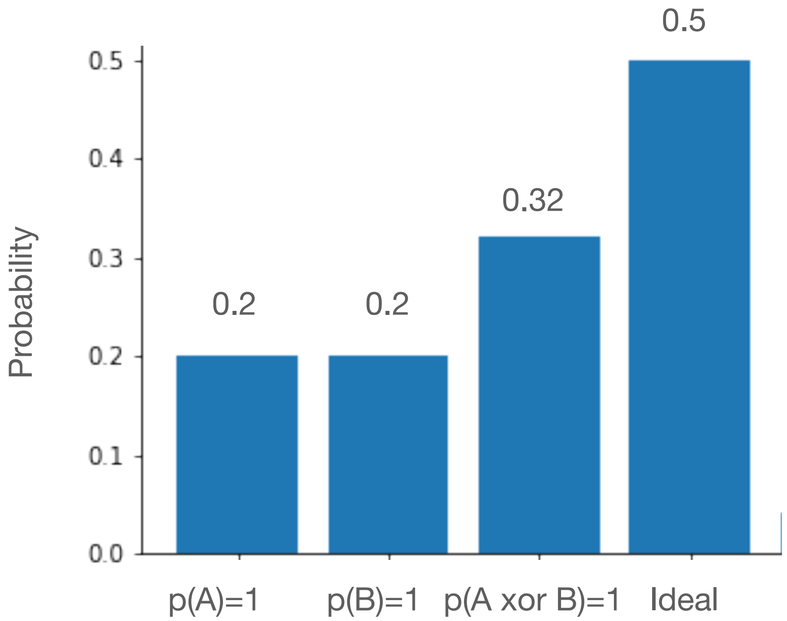


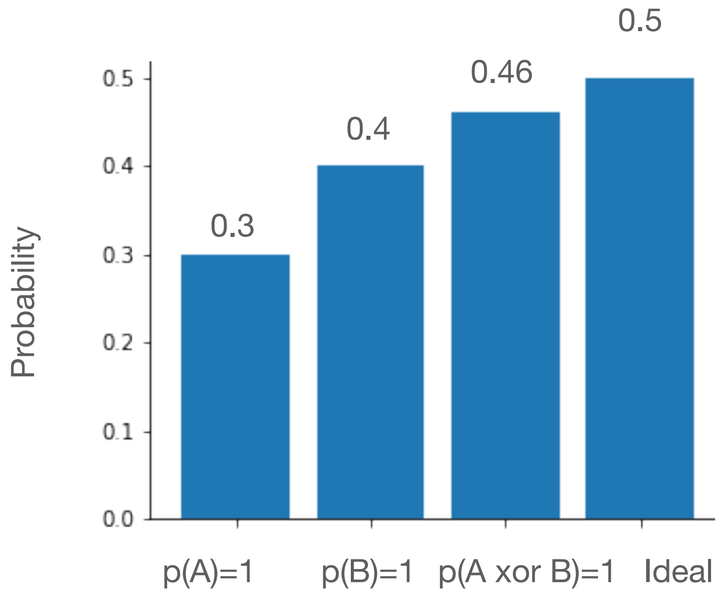




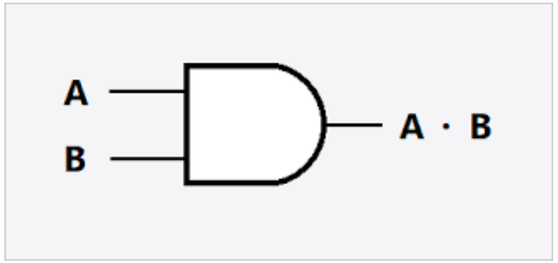
** based on 5000 samples*

Entropy Extractors - a simple extractor

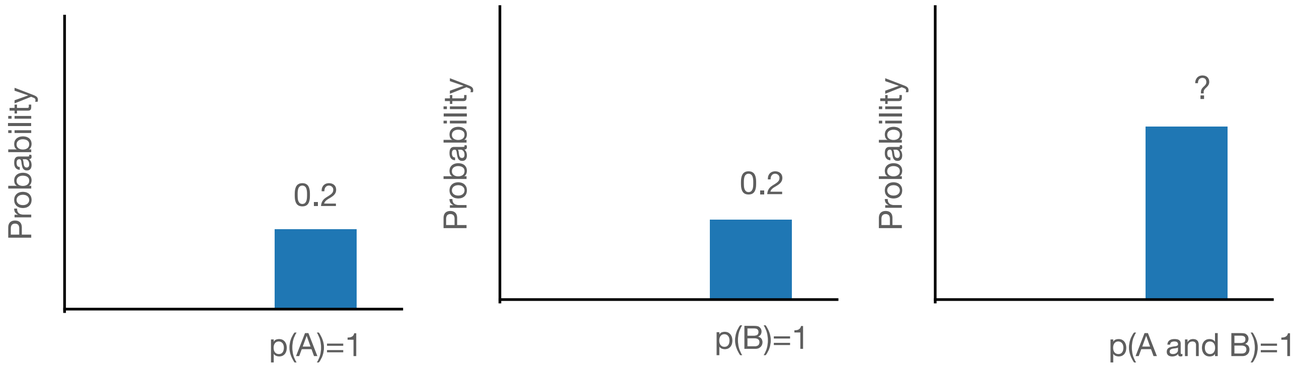




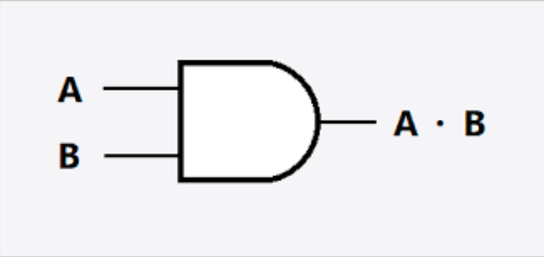
Entropy Extractors - how about AND?



A	B	A and B
0	0	0
1	0	0
0	1	0
1	1	1

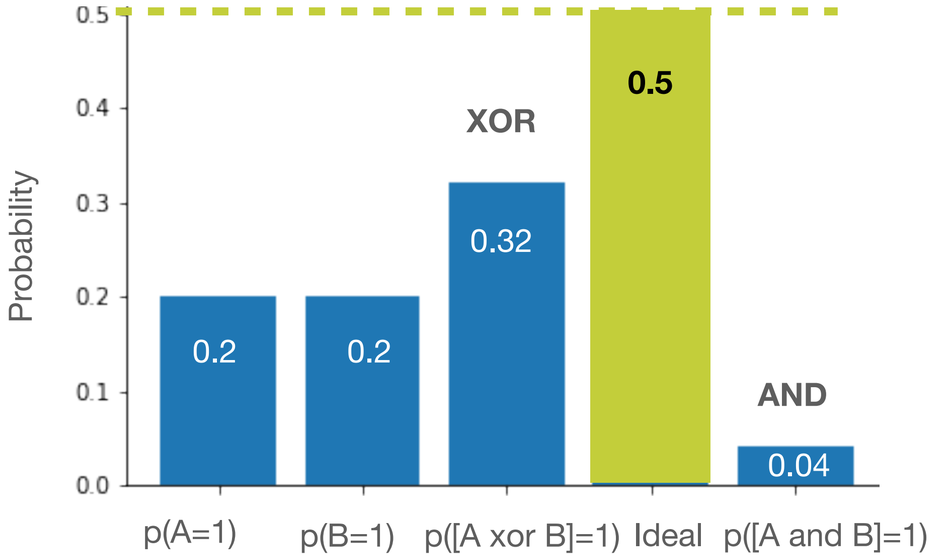


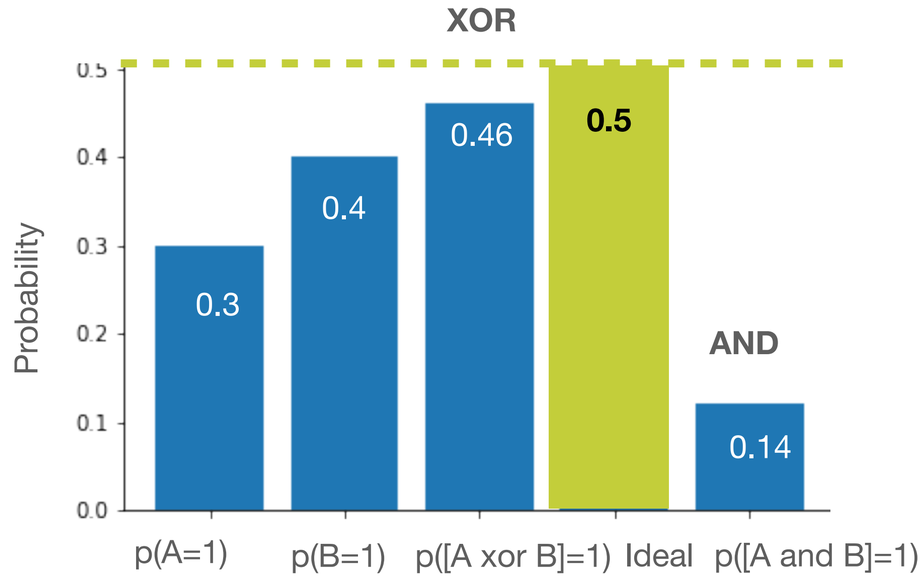
Entropy Extractors - how about AND?



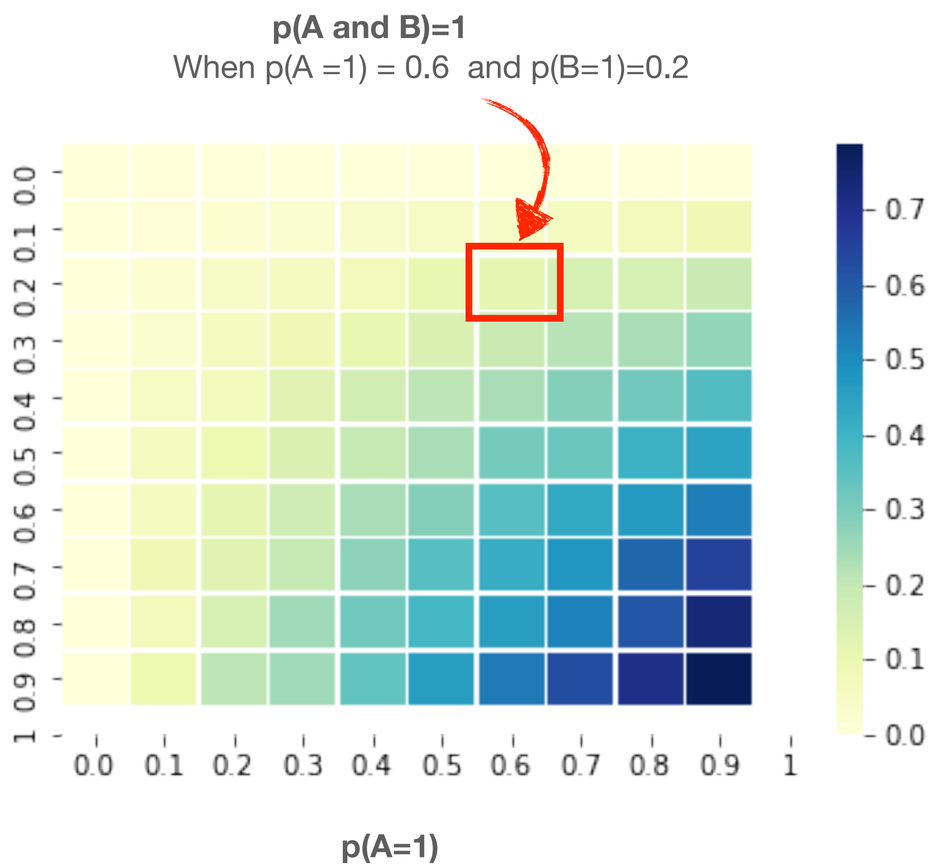
A	B	A and B
0	0	0
1	0	0
0	1	0
1	1	1

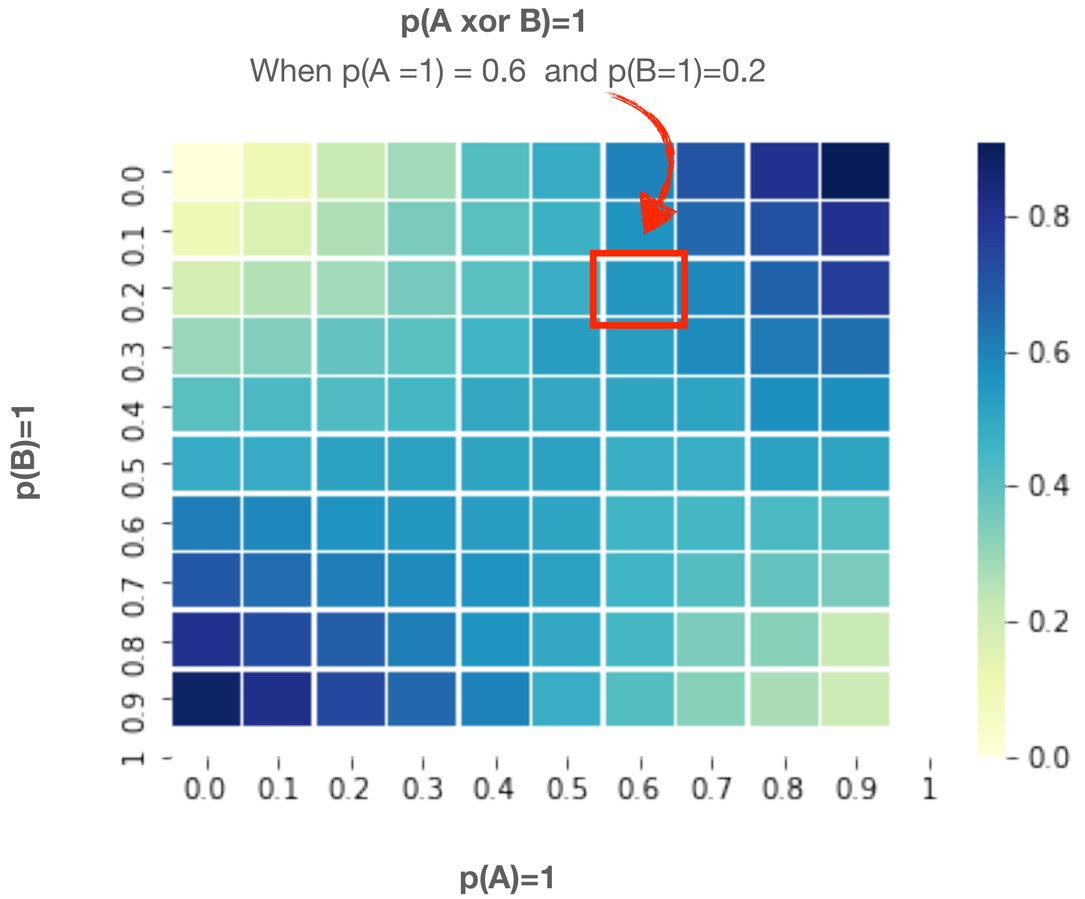
Entropy Extractors - a simple extractor



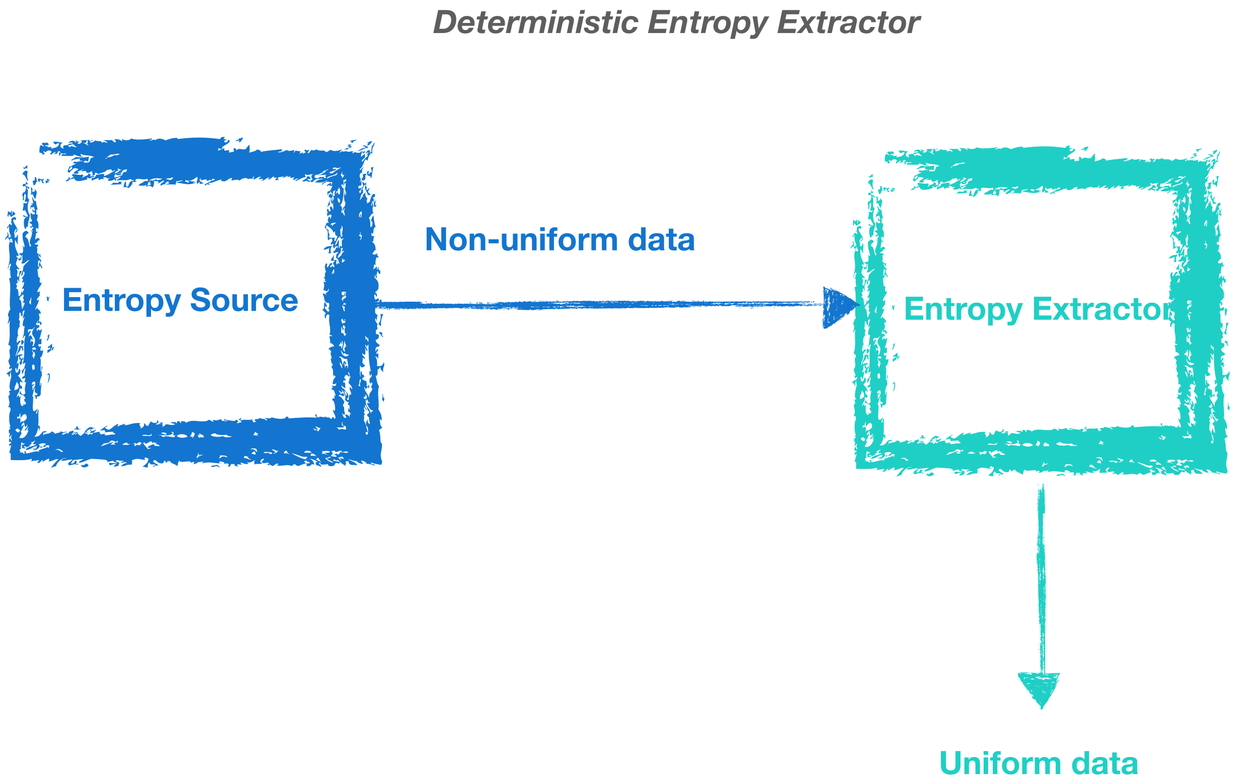


Entropy Extractors





Entropy Extractors - bad news



On the Impossibility of Private Key Cryptography with Weakly Random Keys

James L. McInnes*
University of Toronto

Benny Pinkas†
Technion — Israel Institue of Technology

Abstract

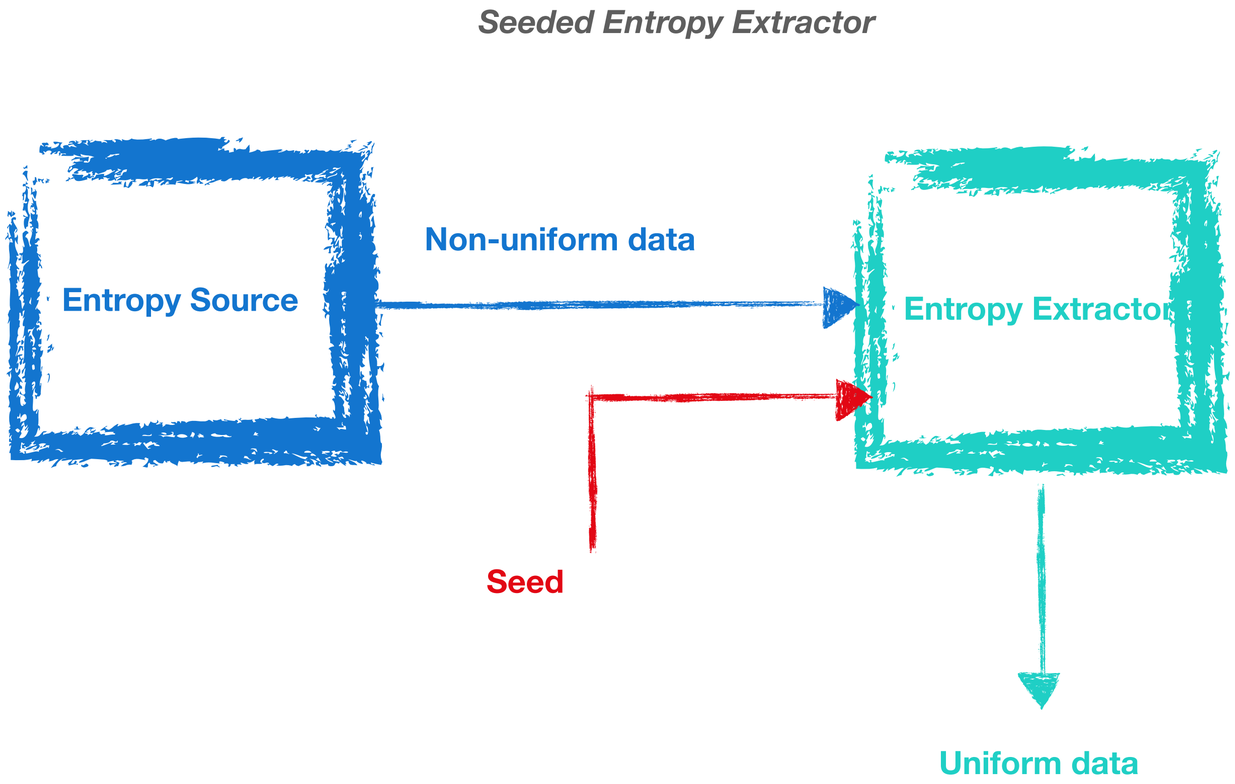
The properties of weak sources of randomness have been investigated in many contexts and using several models of weakly random behaviour. For two such models, developed by Santha and Vazirani, and Chor and Goldreich, it is known that the output from one such source cannot be "compressed" to produce nearly random bits. At the same time, however, a single source is sufficient to solve problems in the randomized complexity classes BPP and RP. It is natural to ask exactly which tasks can be done using a single, weak source of randomness and which cannot. The present work begins to answer this question by establishing that a single weakly random source of either model cannot be used to obtain a secure "one-time-pad" type of cryptosystem.

1 Introduction

Secret transmission of information over insecure communication lines is a major issue in cryptography. In the classical setting, two parties A and B, share a secret, private key K. A wishes to send a plaintext message M, to B. A encrypts M using K, and sends the resulting ciphertext C, to B. A listener L can eavesdrop on the communication line and find C (but not alter it). In addition L knows the functions employed by A and B. The goal of the cryptosystem is to enable B to correctly decrypt M, while retaining security against the listener.

In order to operate, the parties A and B need an access to a joint source of randomness. Without such a source, L possesses the same information as B does. As L knows B's program, B has no advantage over L, and so such a cryptosystem will not be secure.

Entropy Extractors - bad news



On the Impossibility of Private Key Cryptography with Weakly Random Keys

James L. McInnes*
University of Toronto

Benny Pinkas†
Technion — Israel Institue of Technology

Abstract

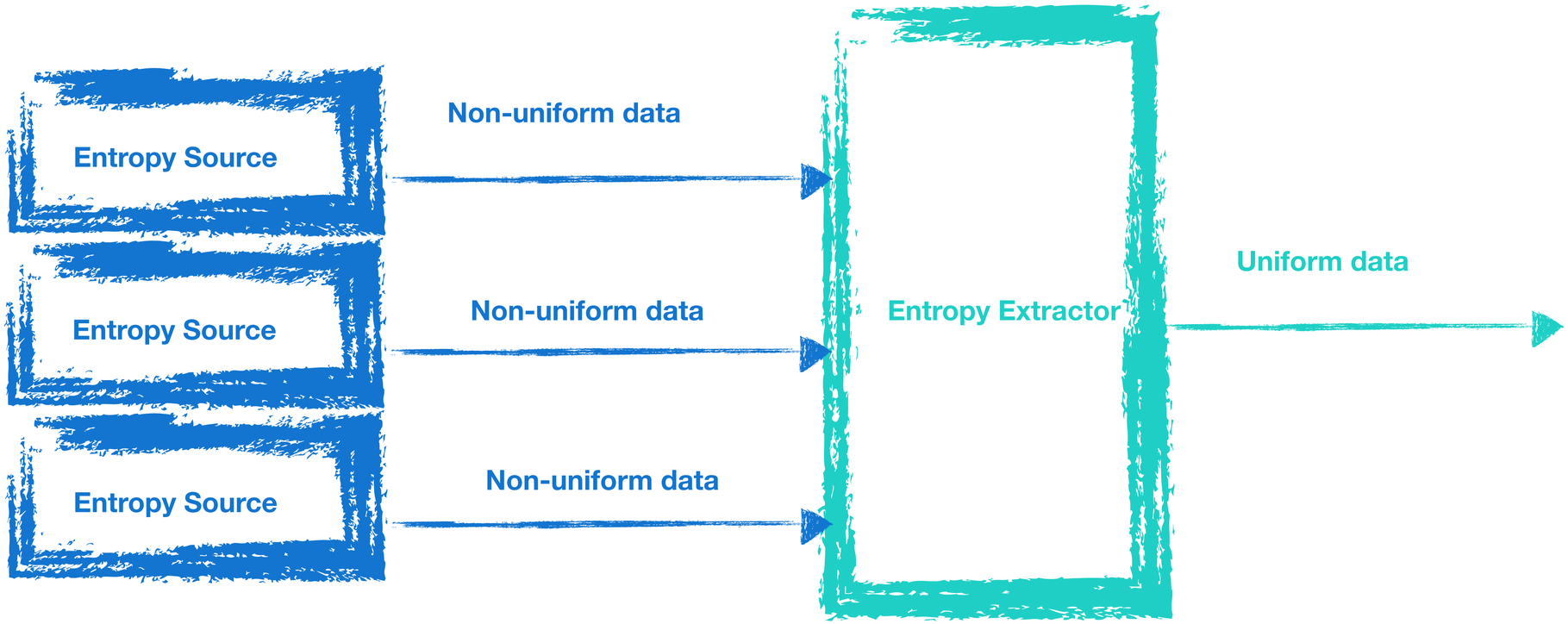
The properties of weak sources of randomness have been investigated in many contexts and using several models of weakly random behaviour. For two such models, developed by Santha and Vazirani, and Chor and Goldreich, it is known that the output from one such source cannot be "compressed" to produce nearly random bits. At the same time, however, a single source is sufficient to solve problems in the randomized complexity classes BPP and RP. It is natural to ask exactly which tasks can be done using a single, weak source of randomness and which cannot. The present work begins to answer this question by establishing that a single weakly random source of either model cannot be used to obtain a secure "one-time-pad" type of cryptosystem.

1 Introduction

Secret transmission of information over insecure communication lines is a major issue in cryptography. In the classical setting, two parties A and B, share a secret, private key K. A wishes to send a plaintext message M, to B. A encrypts M using K, and sends the resulting ciphertext C, to B. A listener L can eavesdrop on the communication line and find C (but not alter it). In addition L knows the functions employed by A and B. The goal of the cryptosystem is to enable B to correctly decrypt M, while retaining security against the listener.

In order to operate, the parties A and B need an access to a joint source of randomness. Without such a source, L possesses the same information as B does. As L knows B's program, B has no advantage over L, and so such a cryptosystem will not be secure.

Mutiple Input Entropy extractors



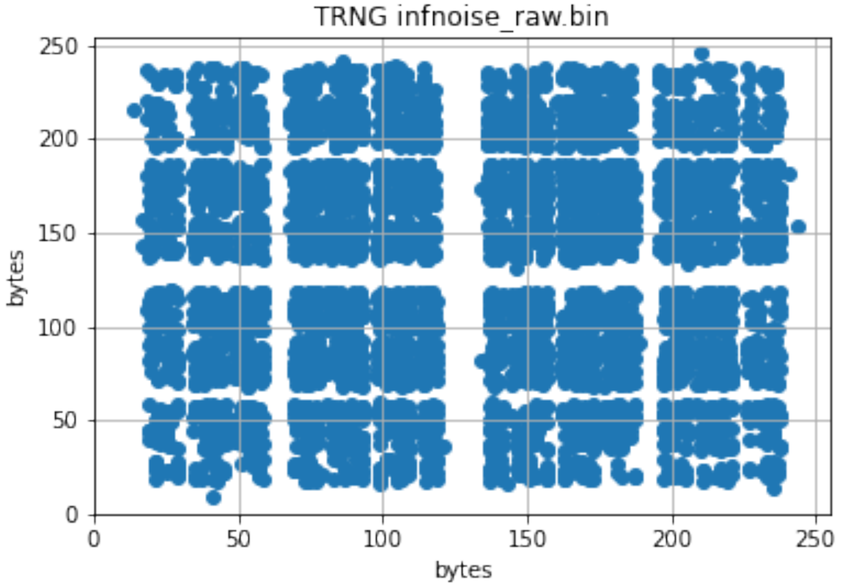
Take away

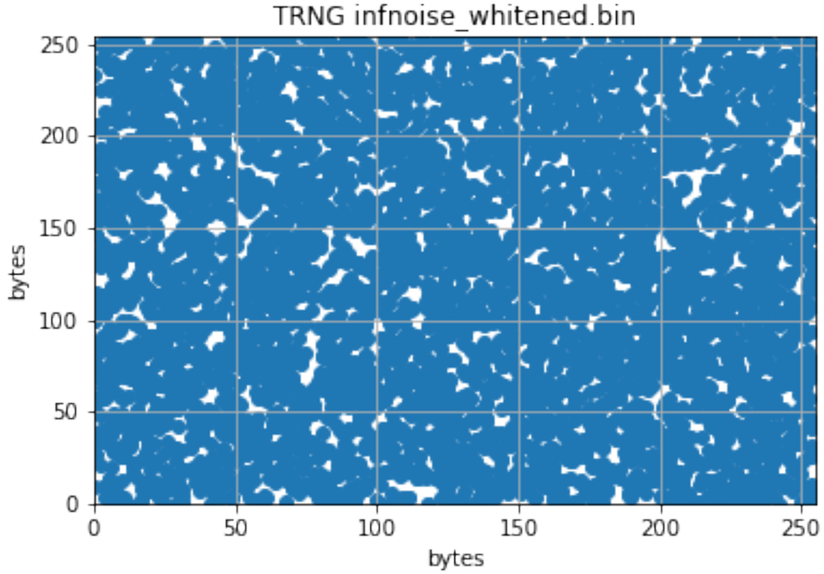
An entropy extractor can transform a non-uniform distribution into a uniform one, but cannot generate more randomness.

How do we evaluate RNGs?

First, we need to know what we want

Does it look random?





There is a better way….

How do we evaluate TRNGs?

- **Uniform random numbers:**
  - Does the output look random?
- **Randomness:**
  - What is the probability of the most likely symbol?
  - What is the average probability of a symbol?
- **Unpredictability (cryptography):**
  - Forward prediction resistance
  - Backtrack resistance
- **Stationary entropy source**
  - Resilient to changes in the environmental conditions/aging

29 49 0d 57 19 0d 18 36 4c 49 54 1e 59 38 01 49 52 41 1f 3c
20 25 34 43 4c 3f 5d 44 3e 05 34 47 4a 5f 5f 46 3e 42 0e 50 3f
0a 25 64 4d 17 3a 5e 19 10 35 04 01 25 3a 0f 43 36 4f 07 25
45 30 d0 c1 82 61 b3 b3 32 d2 74 b5 52 65 42 75 82 41 82
37 39 2c 45 0b 22 16 46 08 26 5a 36 4f 42 3b 35 4e 59 51 4b
64 05 28 5c 3a 1d 19 36 43 06 33 0f 42 09 54 61 4c 2f 02 33
4b 64 02 09 0c 2b 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14 27
23 26 50 5e 45 18 07 34 60 3a 4b 5d 63 5f 3c 45 18 47 0d 14
0c 64 29 36 0c 52 01 2f 37 2e 4f 46 3f 0a 21 4f 0c 27 38 61
2a 33 08 16 17 5b 4b 4d 44 27 20 20 44 2f 5f 4b 15 4a 61 61
4d 17 3a 5e 19 10 35 04 01 25 3a 0f 43 36 4f 07 25
45 30 d0 c1 82 61 b3 b3 32 d2 74 b5 52 65 42 75 82 41 82
37 39 2c 45 0b 22 16 46 08 26 5a 36 4f 42 3b 35 4e 59 51 4b
64 05 28 5c 3a 1d 19 36 43 06 33 0f 42 09 54 61 4c 2f 02 33
4b 64 02 09 0c 2b 5b 48 50 20 18 22 30 35 4f 5e 06 2f 14
23 26 50 5e 45 18 07 34 60 3a 4b 5d 63 5f 3c 45 18 47 00

AIS 20/31 (EUROPE)

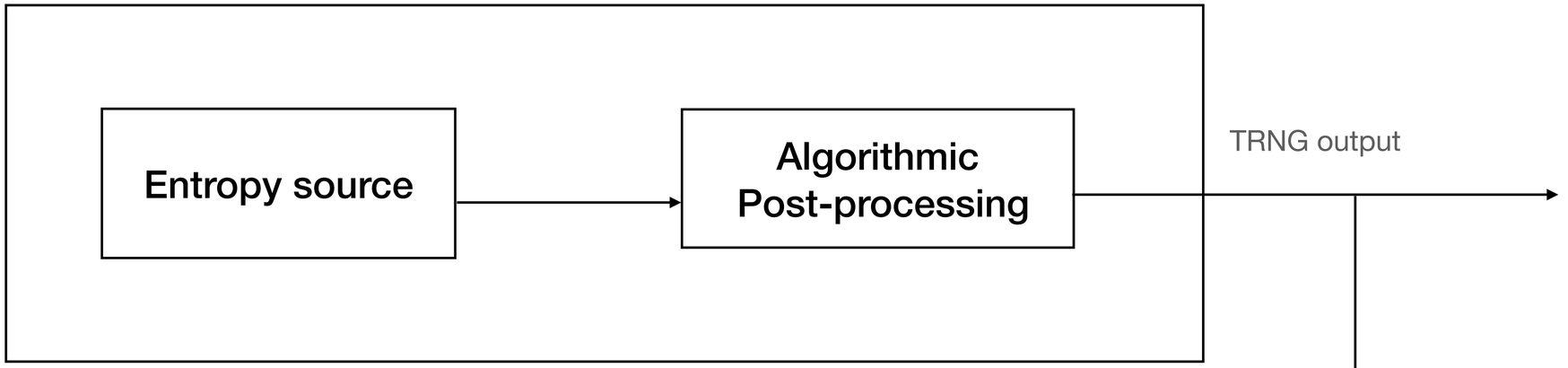
AIS 20/31 (EUROPE)

- German Federal Office for Information Security (BSI) -2001,2011
- Specifies three classes of TRNGs (PTG1-PTG3)

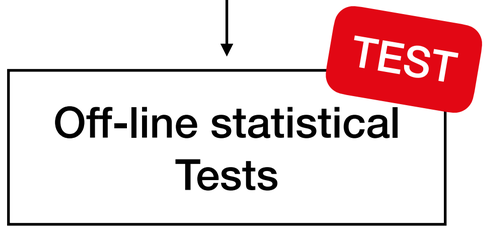
NIST 800-90B (US)

- National Institute of Standards and Technology (NIST)
- Three parts A- PRNG, B-TRNG, C-hybrid, 2006, updated frequently

Traditional TRNG design and evaluation

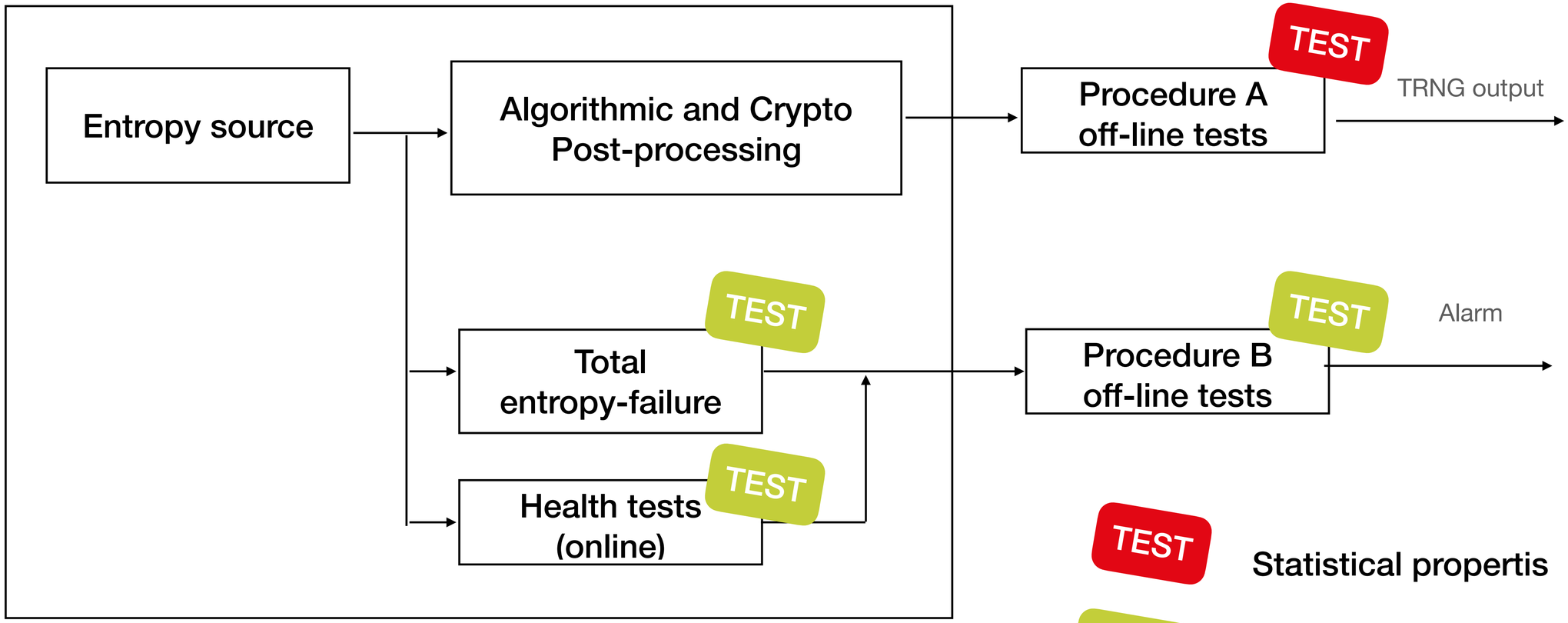


- ☑ TRNG testing same as PRNG
- ☑ Tests look at the generated output and not the entropy source
- ☑ Only simple statistical tests (A)



FIPS 140-1
DIEHARD
NIST 800-22

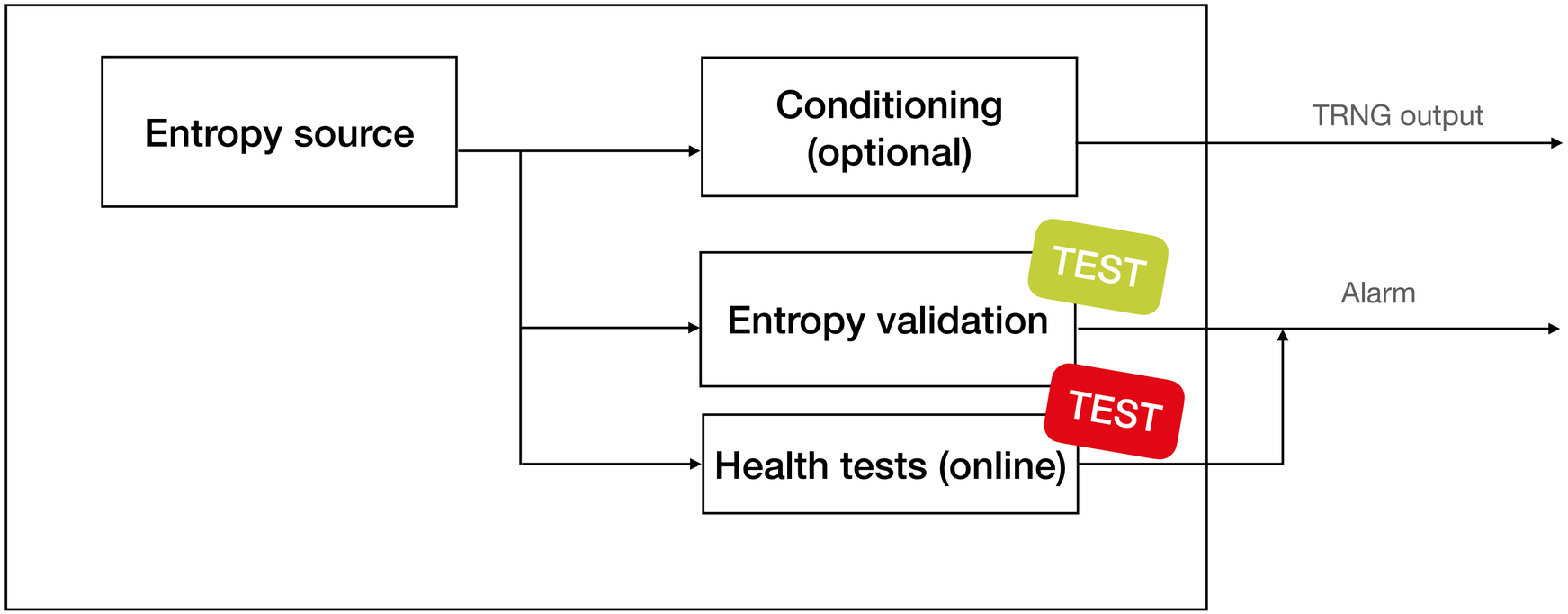
AIS 20/31 (EUROPE)



TEST Statistical propertis

TEST Tests for the raw binary sequence

NIST 800-90B



How do we evaluate RNGs?

Take away

TRNG are technology and design specific
and cannot be standardized.