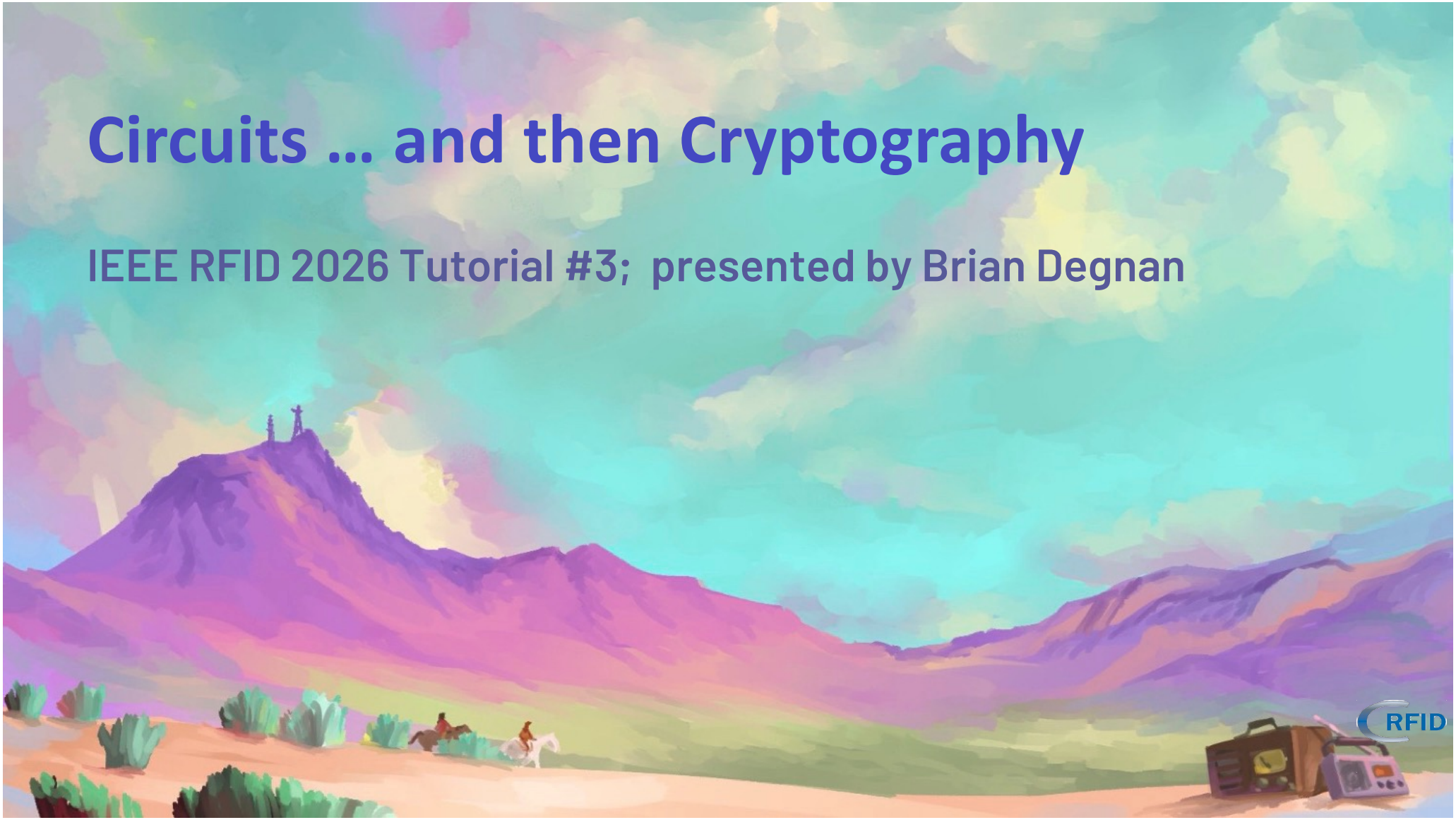
Circuits … and then Cryptography

IEEE RFID 2026 Tutorial #3; presented by Brian Degnan



Purposes of this Tutorial

- Dynamics of choice from a circuit's perspective
- The costs related with choices
- Digital (mostly) system that is the tag IC
- Gen2V3, what's changed
- Cryptography lite

Outline

- IC design
  - How small, components, charge pumps, frequency
- Data
  - Appearance, protocols, random numbers, demodulation
- Talking back
  - Tag state machine, Query command, slot counter, collision/recovery
- Collisions and Randomness
  - Collisions, RN16, uses of random numbers
- Cryptography
  - Goals, lingo, symmetric examples Gen2V2, updates in Gen2V3

Who made this possible?

The Team at IC Analytica!

Past supporters:

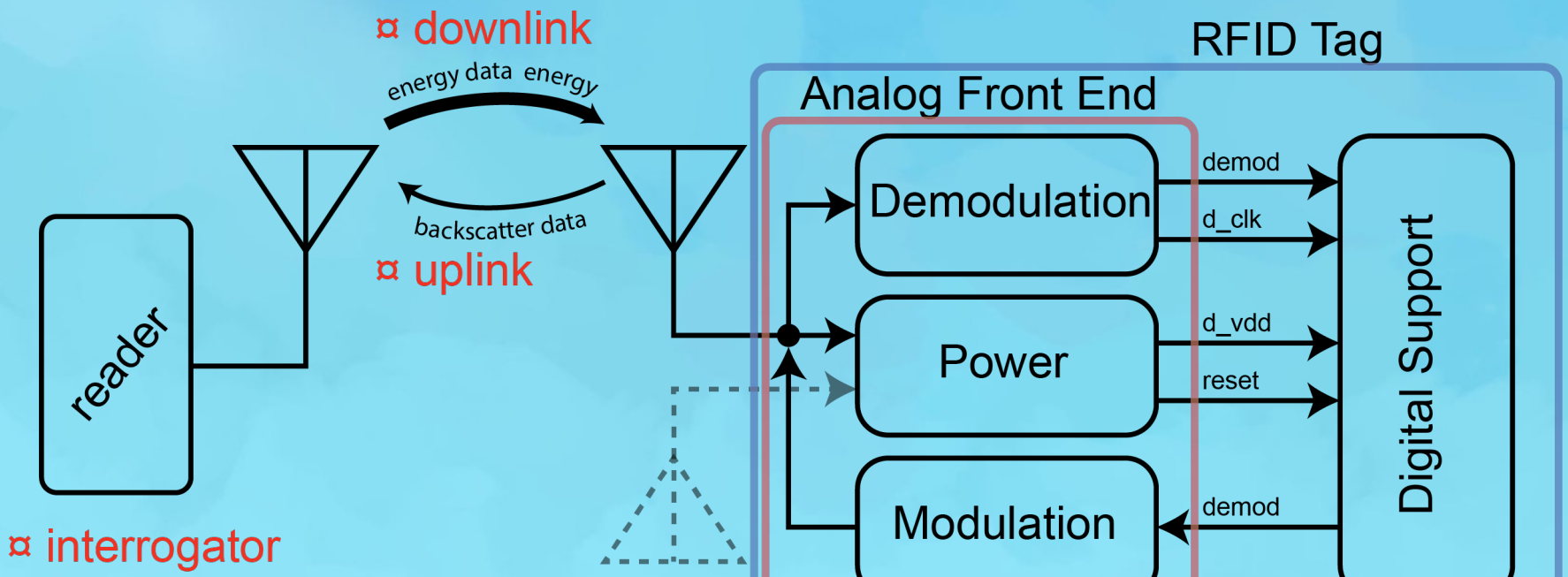
Georgia Institute of Technology

white river technologies

IC Postdoctoral Research Fellowship Program

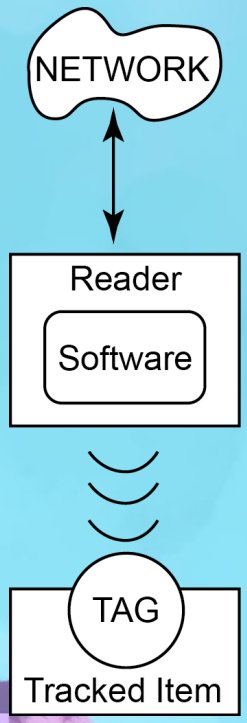
Research Institutes · National Laboratories · Universities

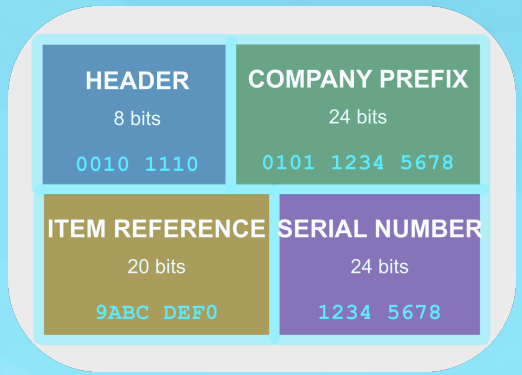
The RFID System: It's digital (mostly)!



- Passively Powered
- Gen2: 860MHz-960MHz +- 40kHz-640kHz

System Overview EPC gen2v2 in context…

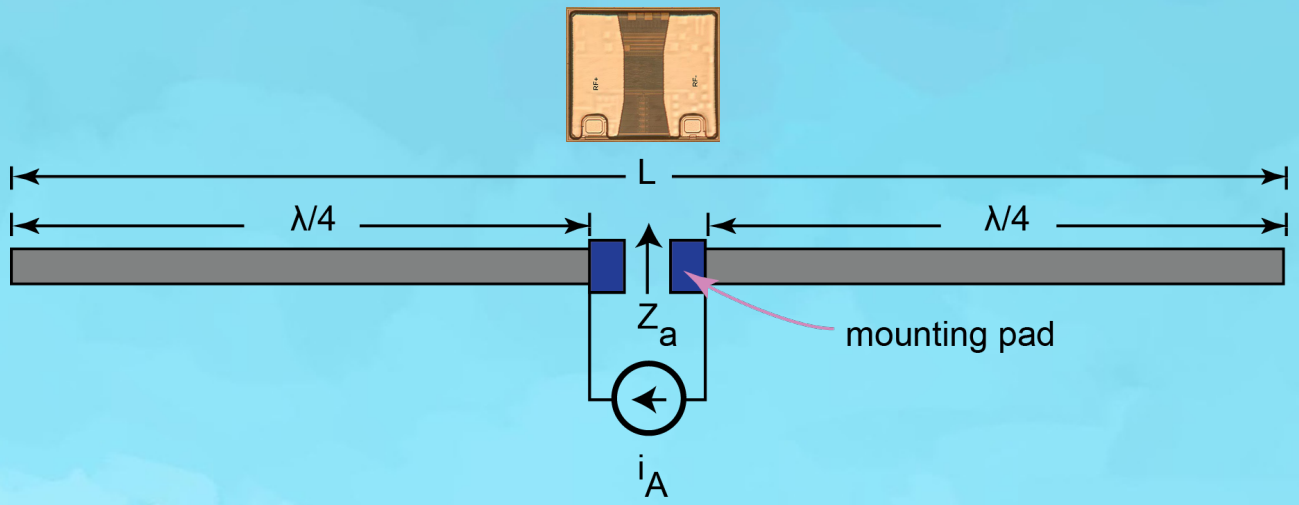




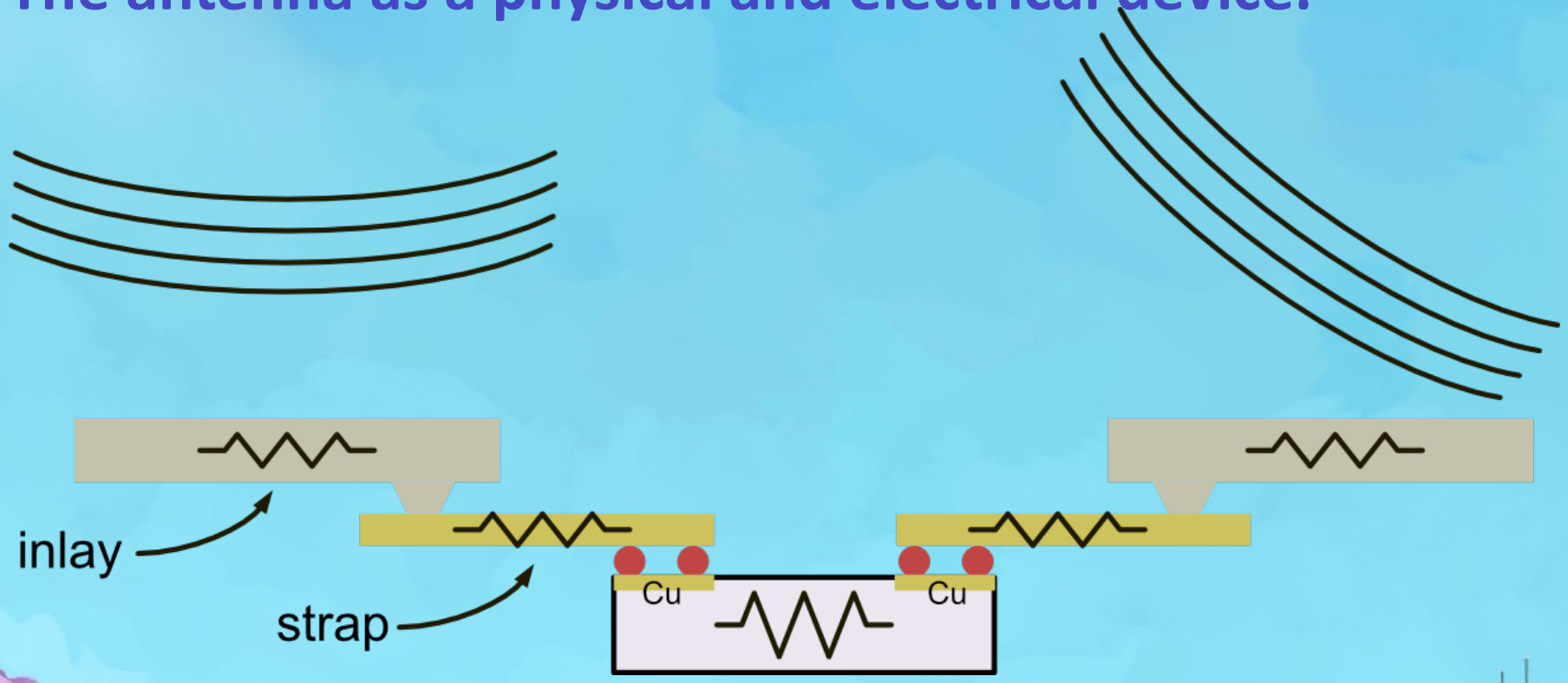
https://www.gs1.org/standards/rfid/uhf-air-interface-protocol

Digital Systems: With a cost driver twist

- How cheap can we make? And what does this mean?
- What drives minimum size? IC or inlay?



The antenna as a physical and electrical device.



Let's look at the cost breakdown (we're at rock bottom).

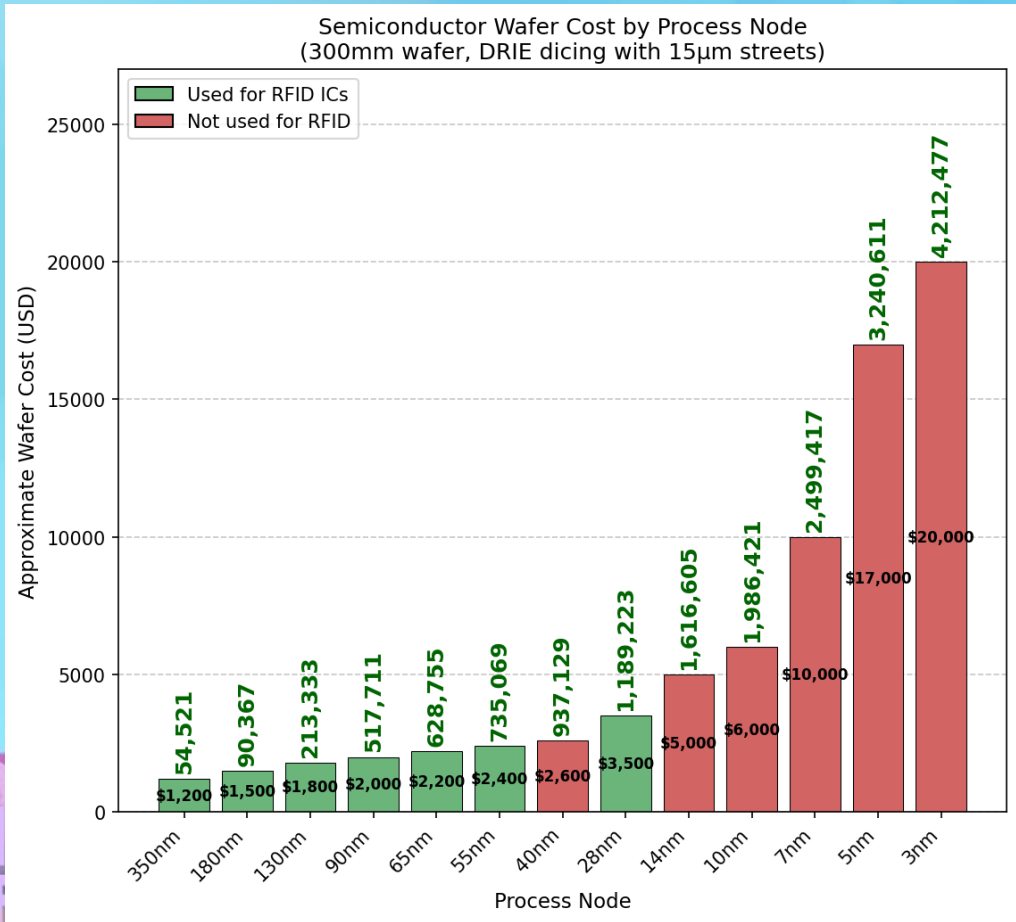
Table 1: Estimated Cost for a Monza 6-Class UHF RFID Inlay, 10^8 units

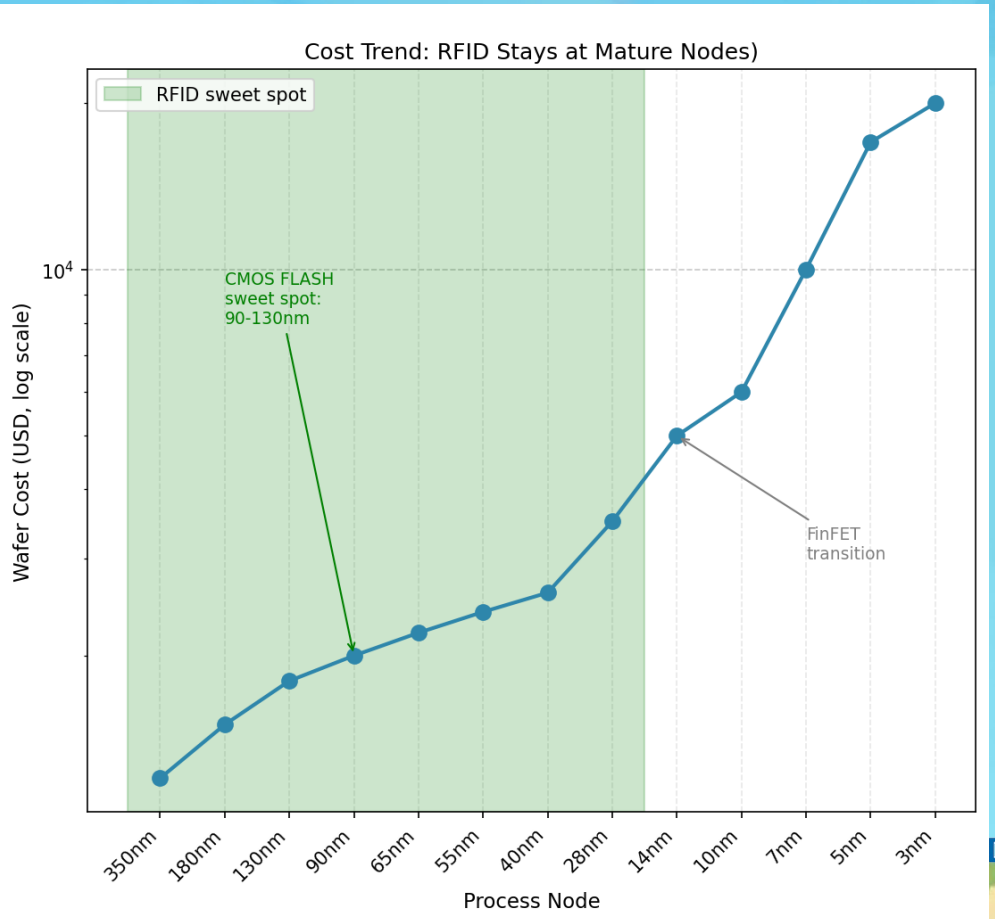
Component	Cost Share (%)	Cost per Inlay (USD)	Die (%) Normal	Notes
Antenna (Al on PET)	40%	0.0200	1.6	Etched aluminum foil + PET substrate
RFID IC (Monza 6)	25%	0.0125	1.0	~0.19–0.21 mm^2 die, wafer sort included
Strap / Interposer	12%	0.0060	0.48	Copper strap fabrication and plating
Assembly	13%	0.0065	0.52	Pick-and-place, adhesive, curing
Yield loss, scrap, test	10%	0.0050	0.40	Wafer yield + inlay fallout
Total	100%	0.0500	4.0	

- ½ wavelength are the cheapest antenna inlays:

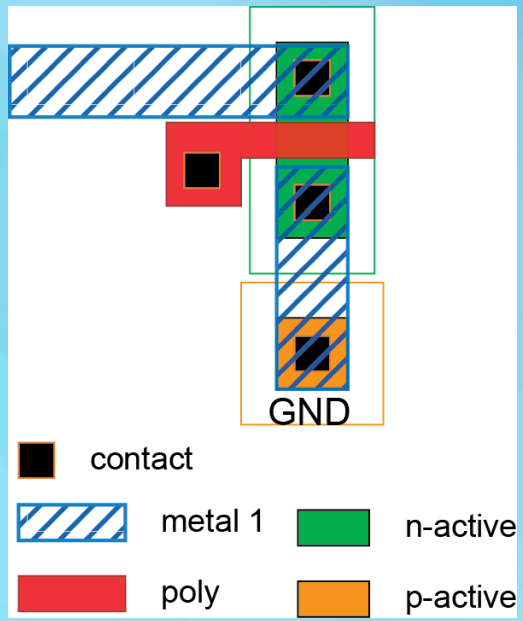
Economies of scale, Design complexity, matching networks
Yield and etch, substrates, etc.

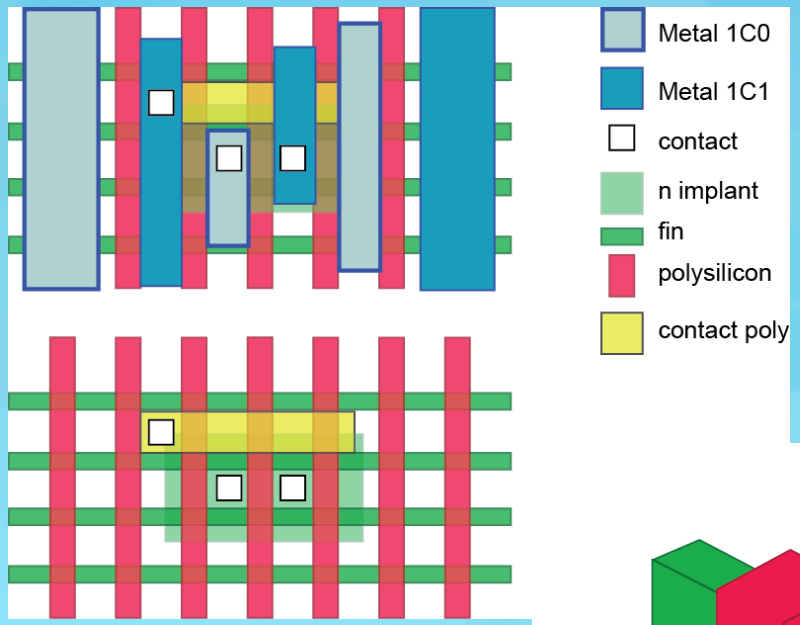
Estimated Die costs (lit. review)

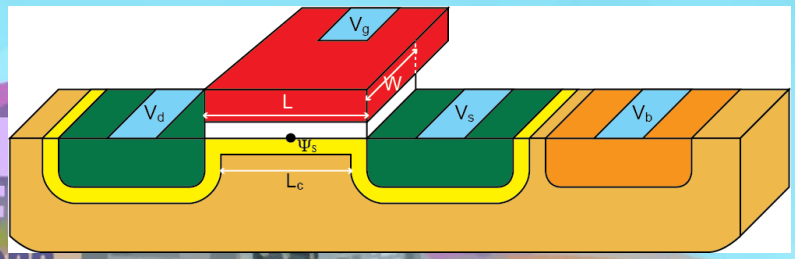


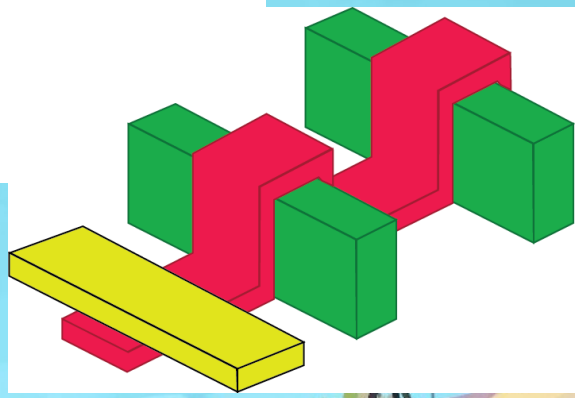


Party like it's 1999… not 2026

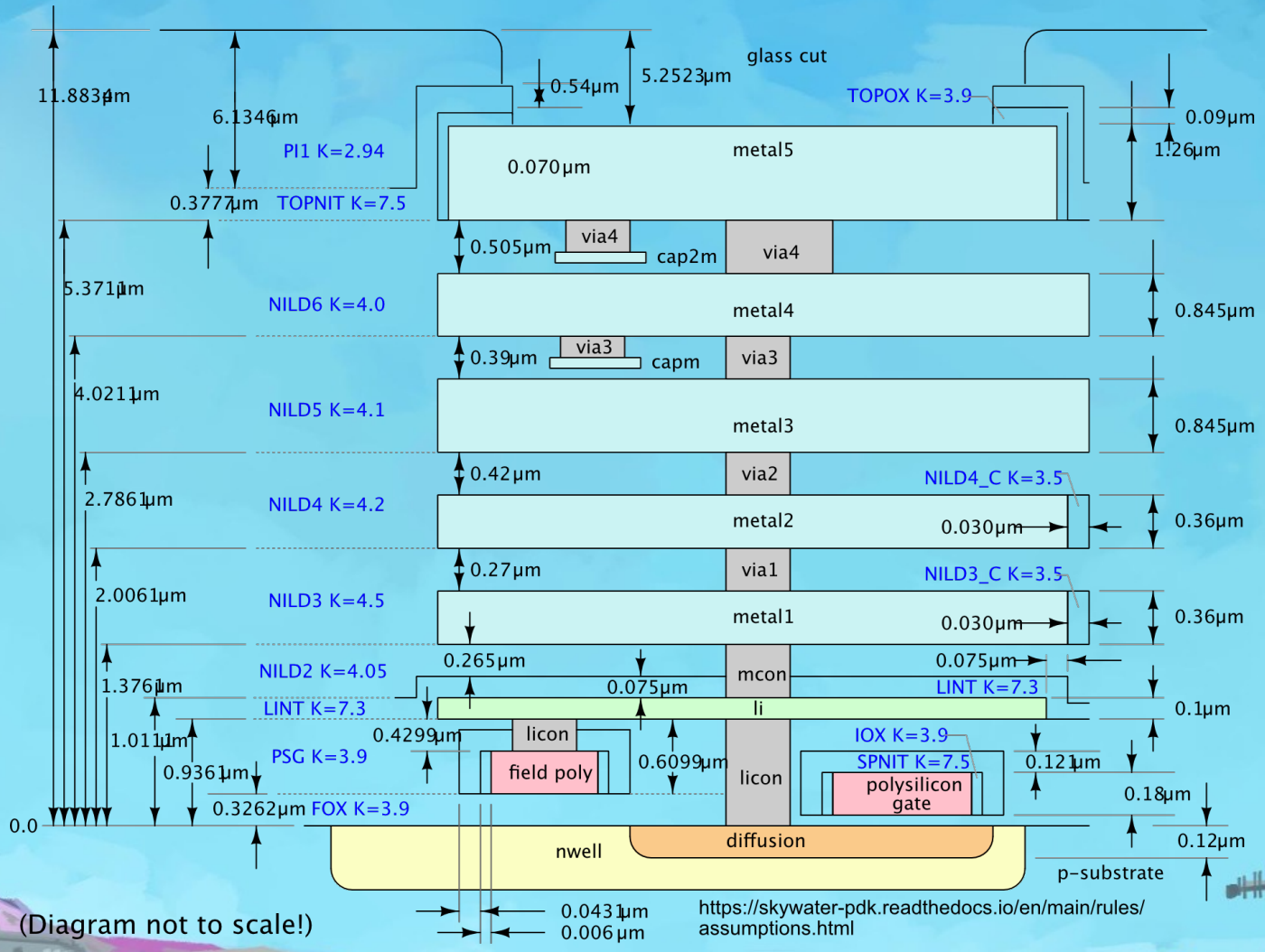








Now on to 130nm...
SKYWATER130



(Diagram not to scale!)

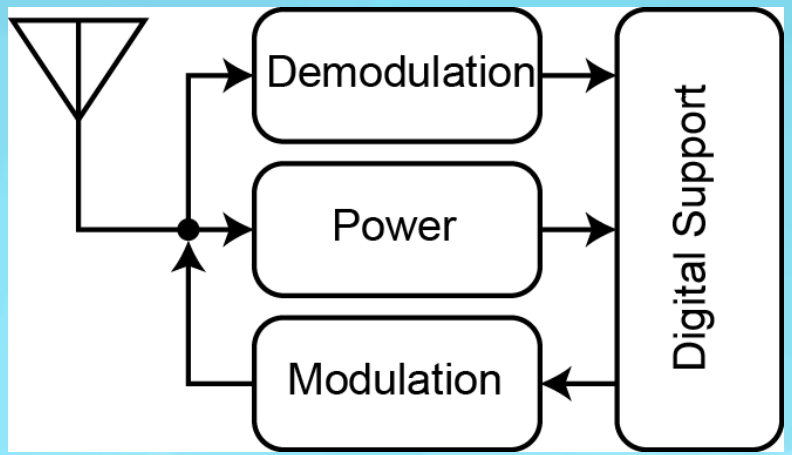
https://skywater-pdk.readthedocs.io/en/main/rules/assumptions.html

Unique things about 900MHz RFID

- Coherent clocks
  - --The RFID tag does not need an oscillator at 900Mhz
- FM0, Reply not FM0
  - -- Reply type is selectable
- Can be passively powered?
  - -- If you can turn it on, and keep it on
    - ...and you have enough power at range

System Punch List

- Harvest Energy (carrier wave supplied)
- Create a stable power supply (use the energy)
- Reset the System (ready the system)
- Decode the incoming data stream (extract)
- Respond to the decoded data (return)



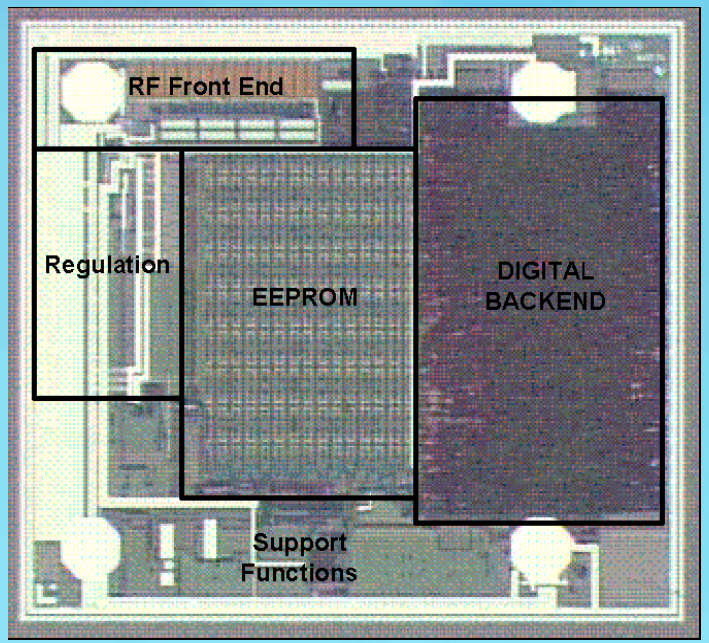
Example Tag (TI with data, Monza with guesses)

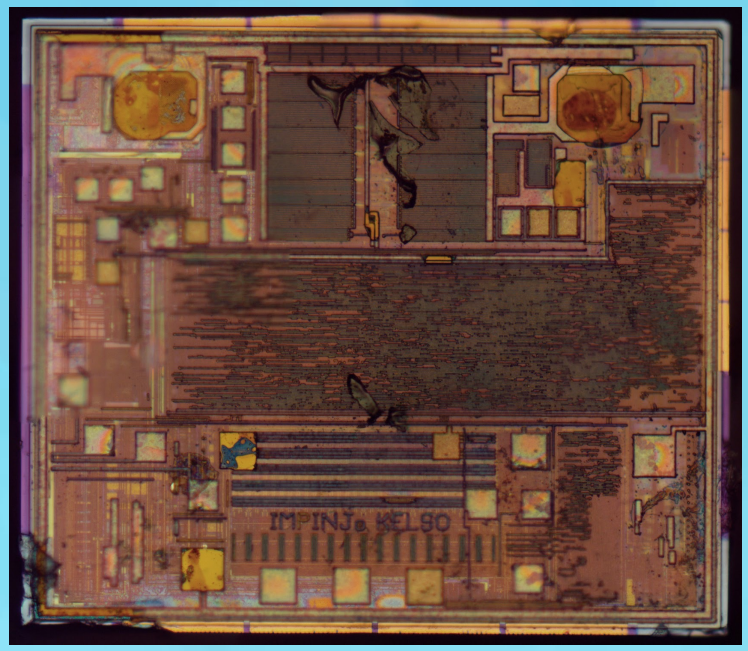
0.5mm^2 Tag:

Digital ~ 30%
EEPROM ~ 20%

RF + DC reg ~ 20%

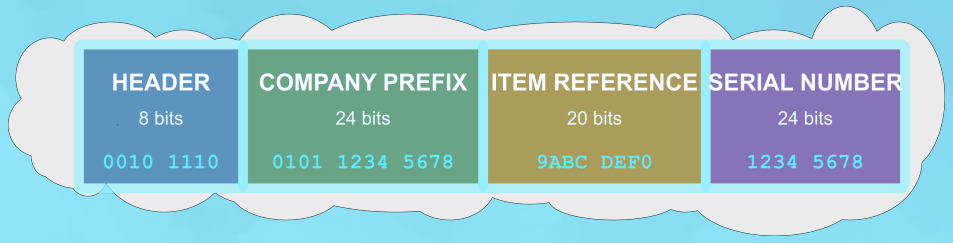
Others (RNG, Charge Pump, support functions): 30%

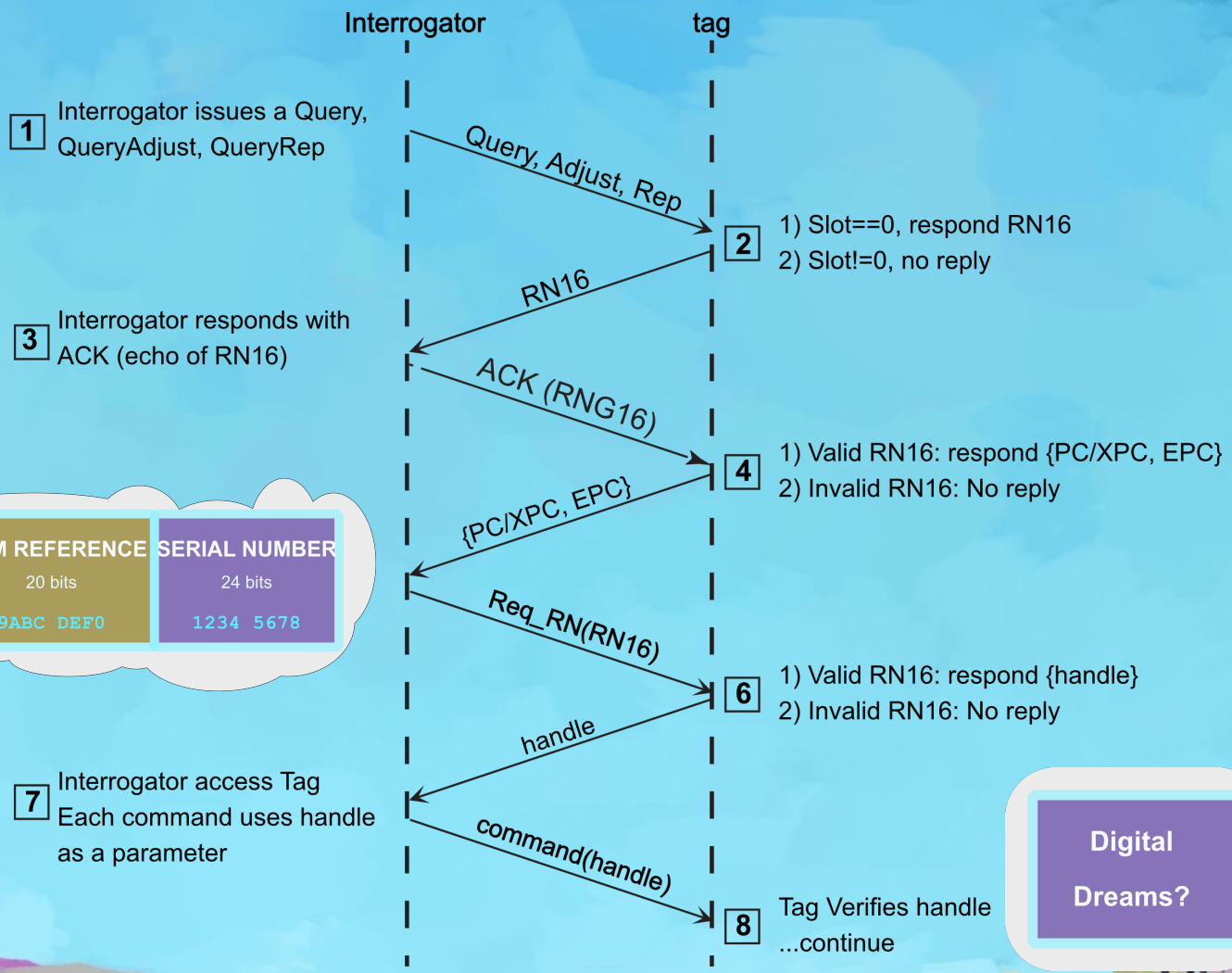




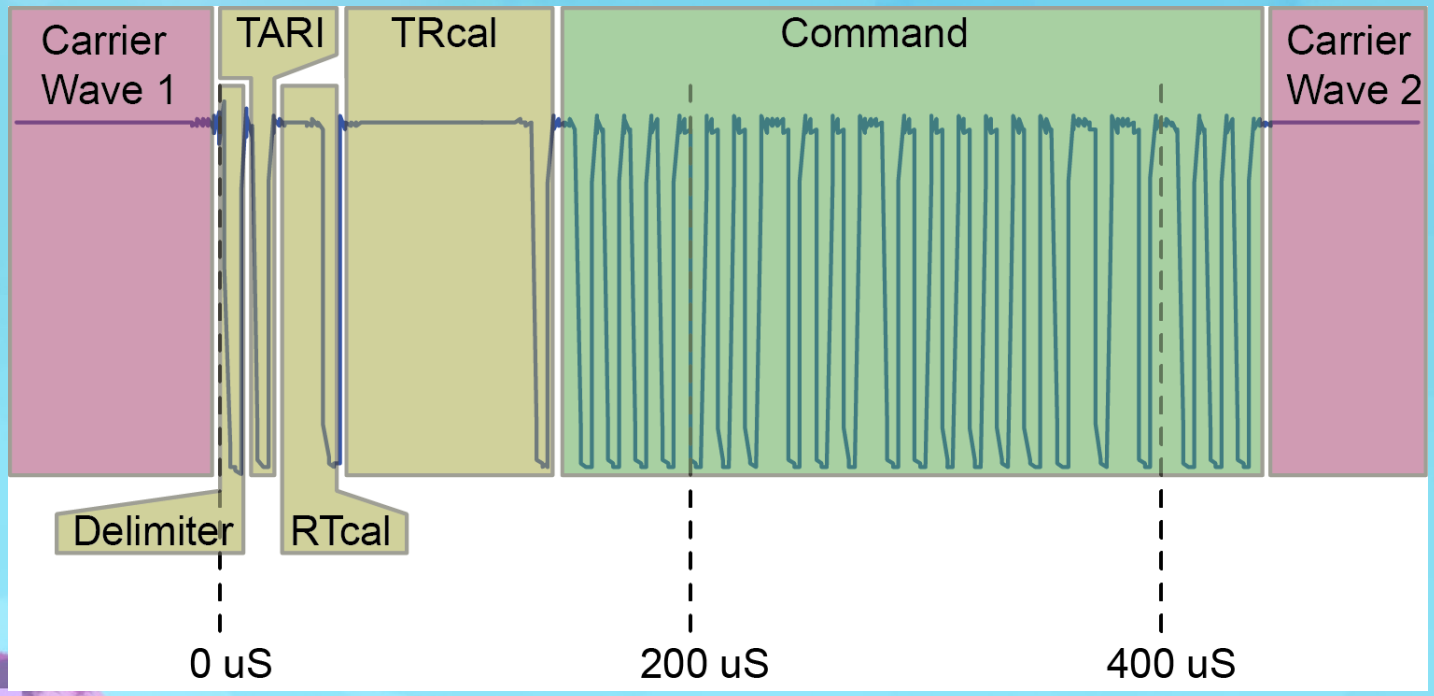
Barnet et al, "A Passive UHF RFID transponder for EPC Gen2 in 0.13um CMOS" (TI's gen2 tag) ISSCC 2007

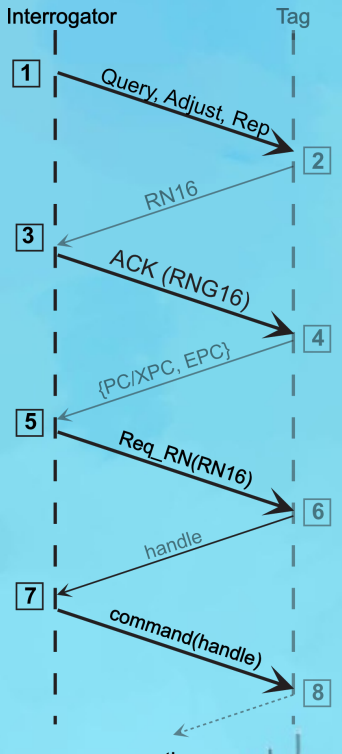
Gen2 Protocol



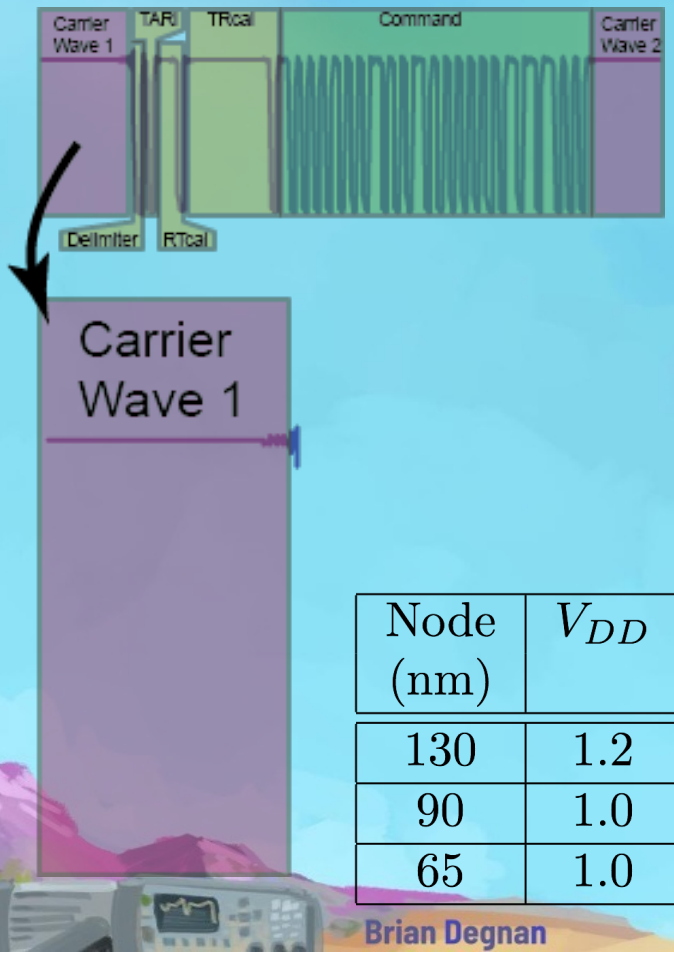


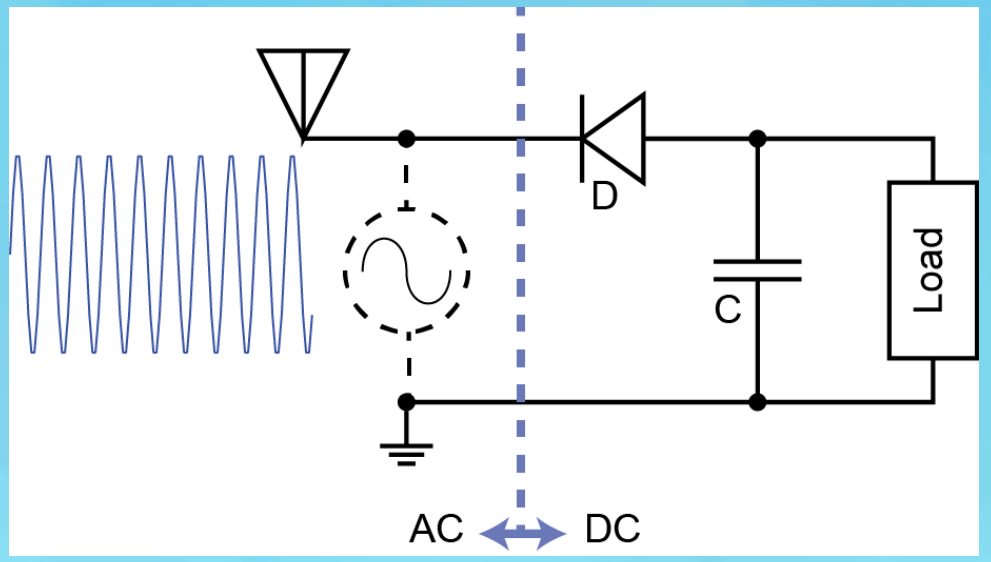
Gen2 Waveform and Vocabulary (900MHz-ish) (After demodulation)





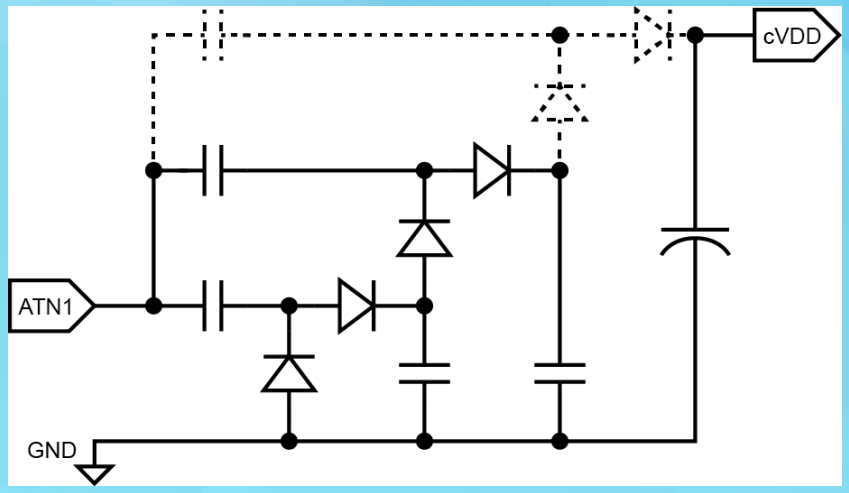
Energy Harvesting: It's a huge load of….everything.

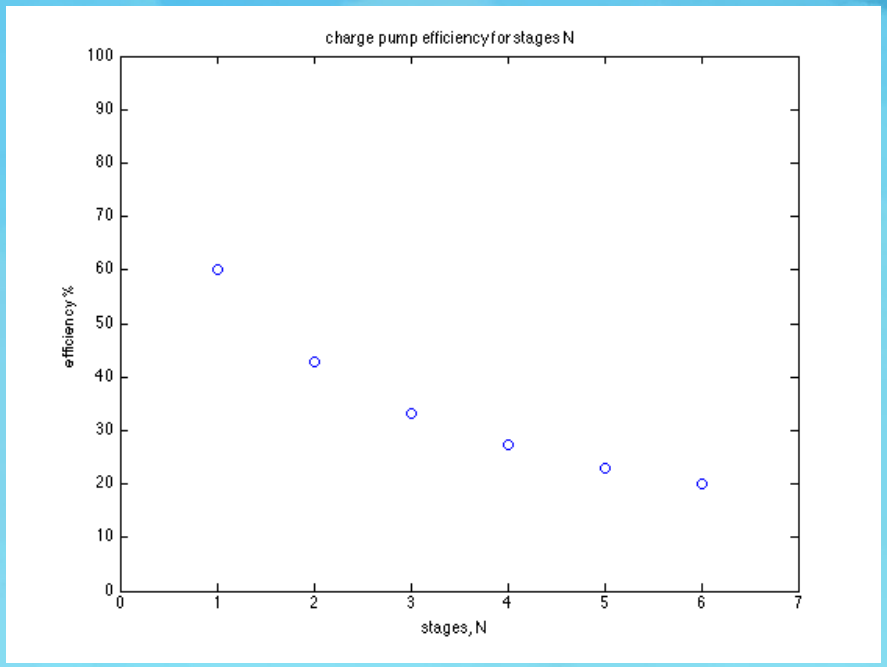




Node (nm)	V_{DD}	I_{sat} (μA)	V_{thn}	Sub-Vt%	I_{DS0} (pA)	ITRS target $I_{DS0}(pA)$
130	1.2	55.5	0.4	33.3	167.1	130
90	1.0	44.4	0.38	38.0	960.3	900
65	1.0	48.5	0.4	40.0	8138.7	6500

Energy Harvesting



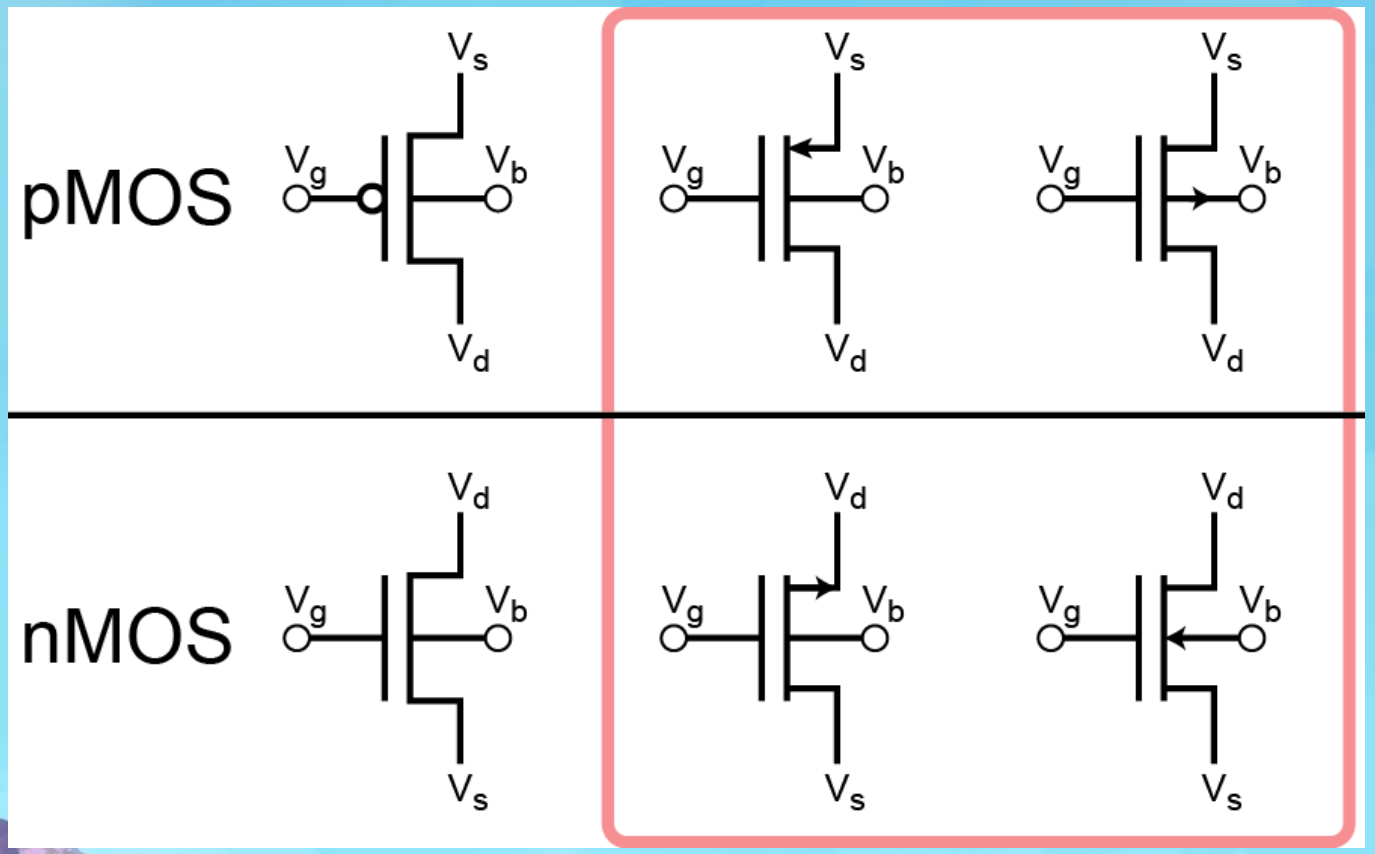


- Number of stages
- Received Energy
- Current Load
- Et Cetera

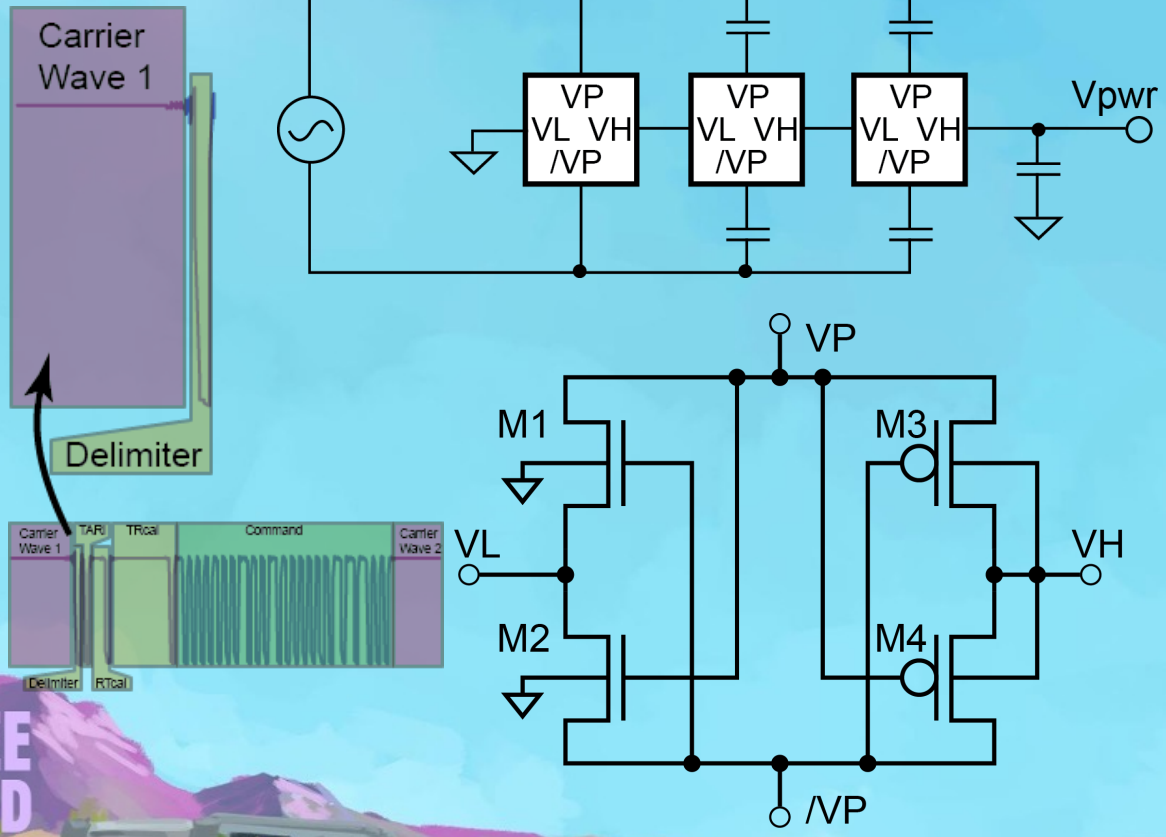
$$V_{DD_{id}} = 2N\left(V_p - V_{on}\right)$$

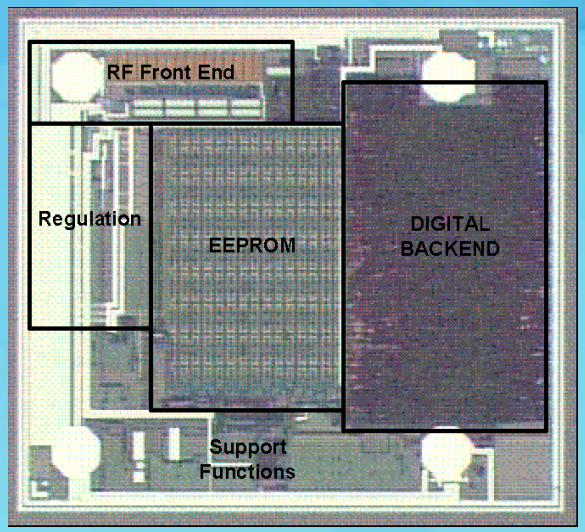
$$\eta_{cp} = \frac{V_{DD}}{V_{DD} + 2NV_{on}}$$

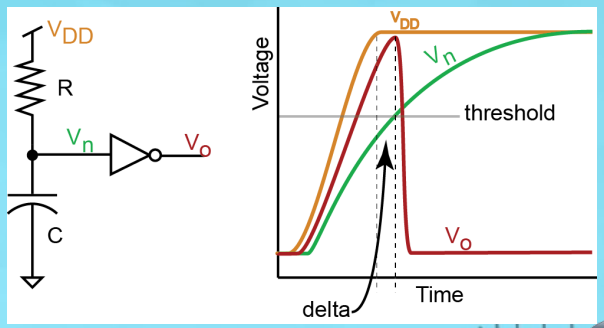
Transistor Symbols



CMOS H-Bridge Charge Pump







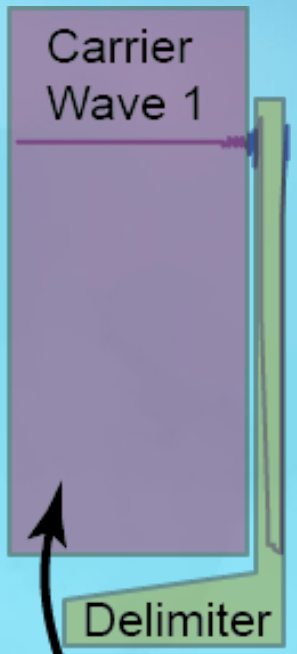
IEEE RFID 2026

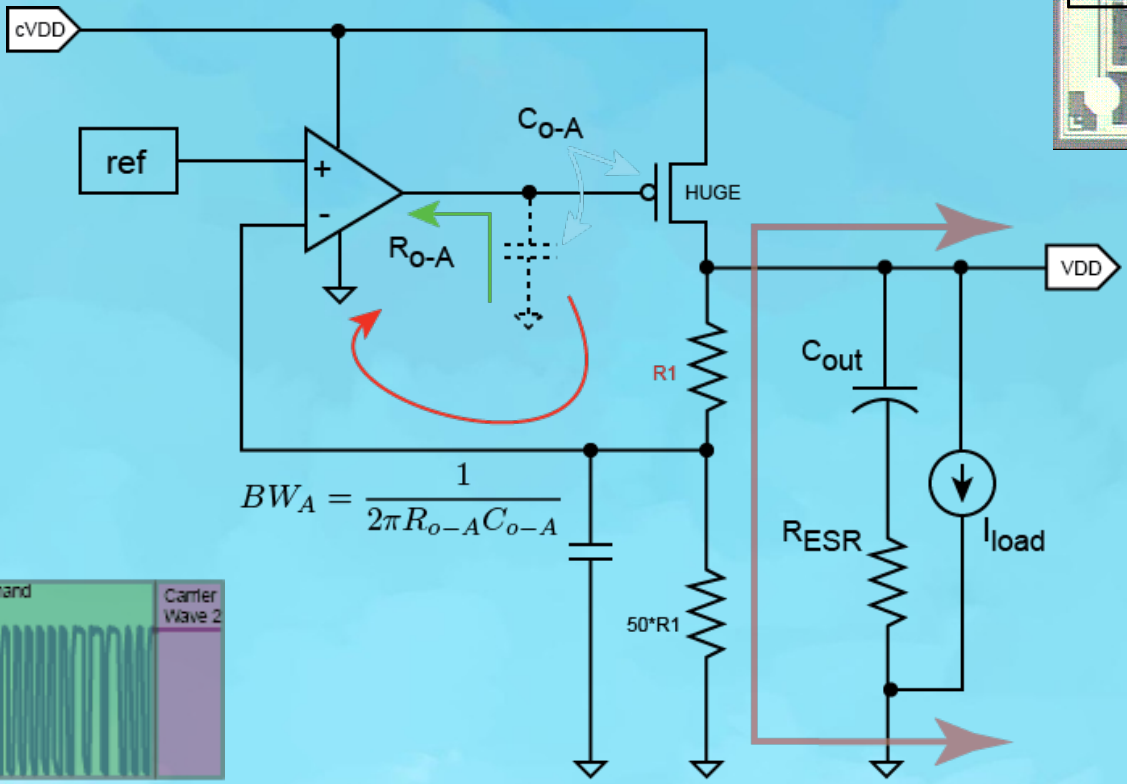
Brian Degnan

RFID

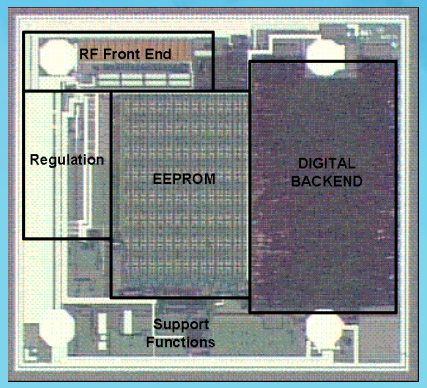
Circuits & Cryptography

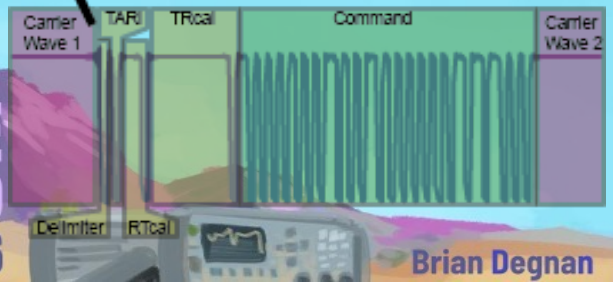
Power Regulator



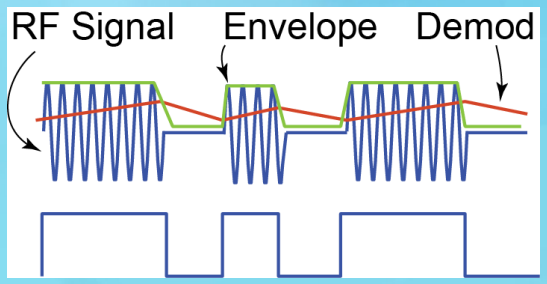


$$BW_A = \frac{1}{2\pi R_{o-A} C_{o-A}}$$

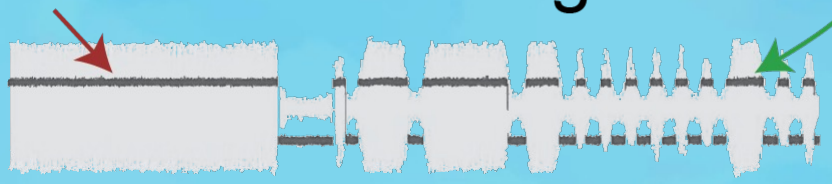


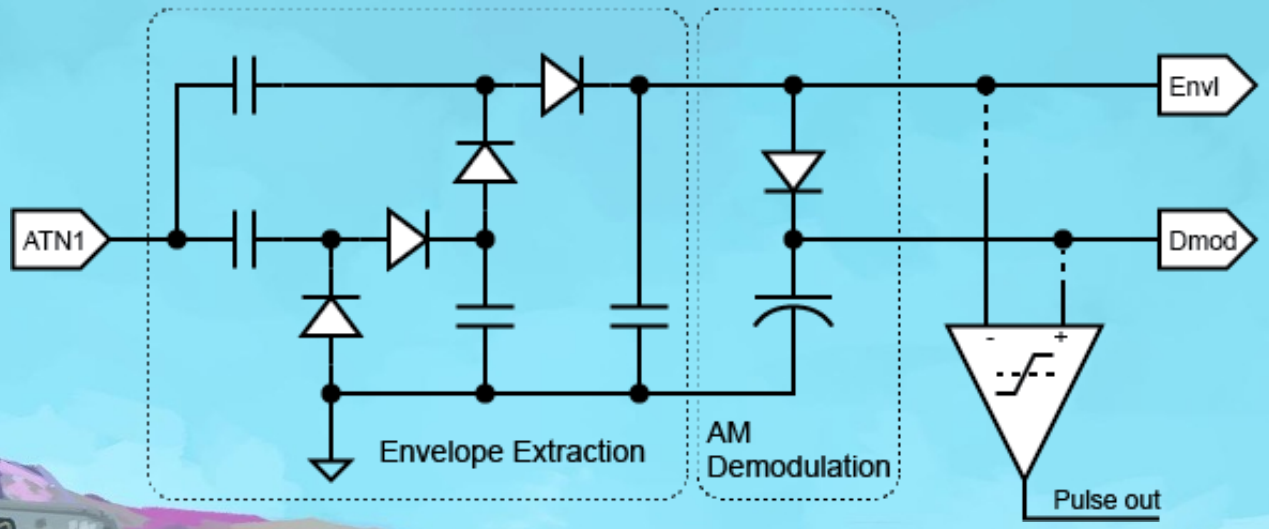


Demodulation: It's just AM.

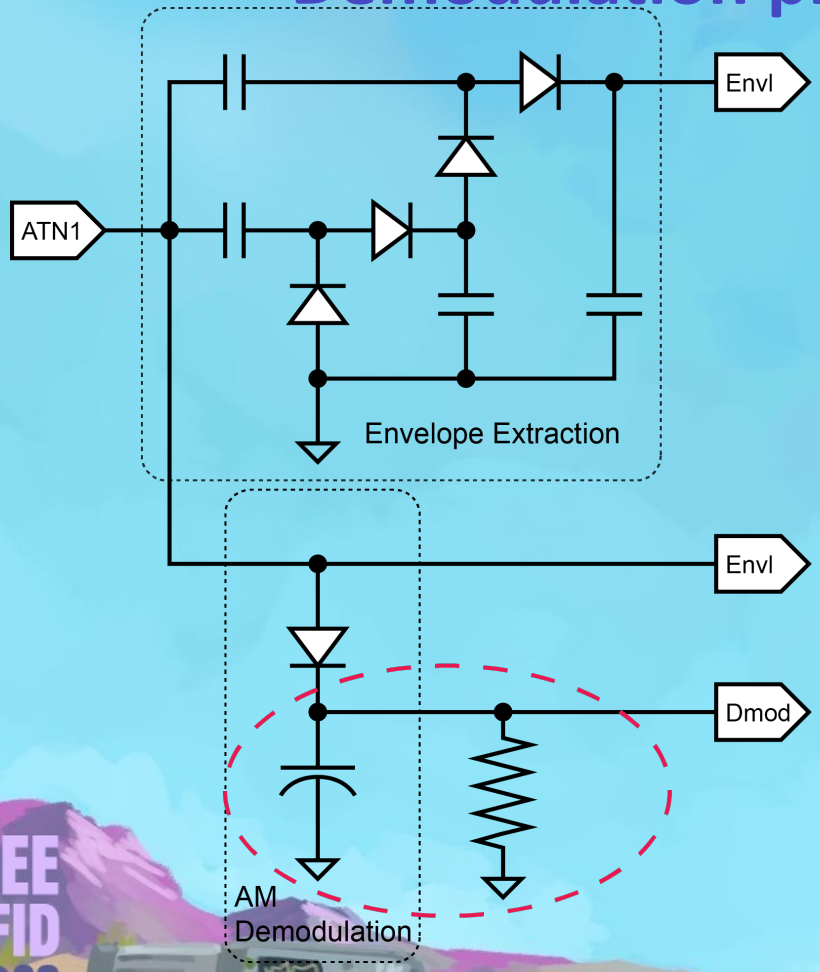


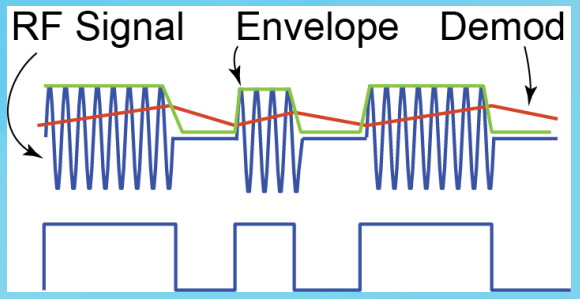
RF waveform Digital Information





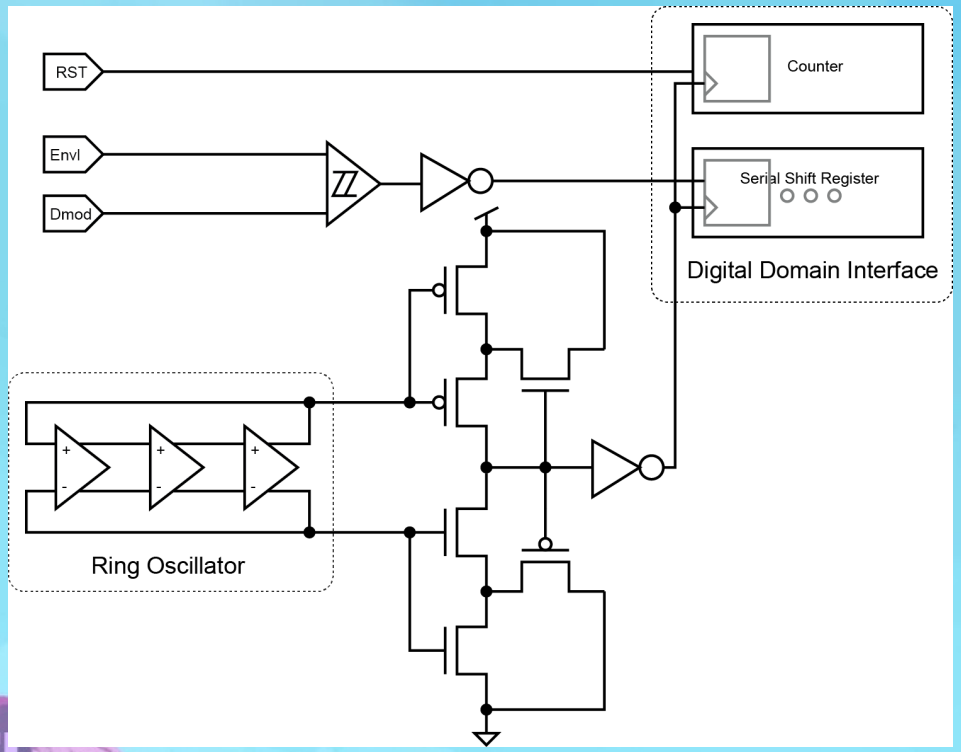
Demodulation pitfalls

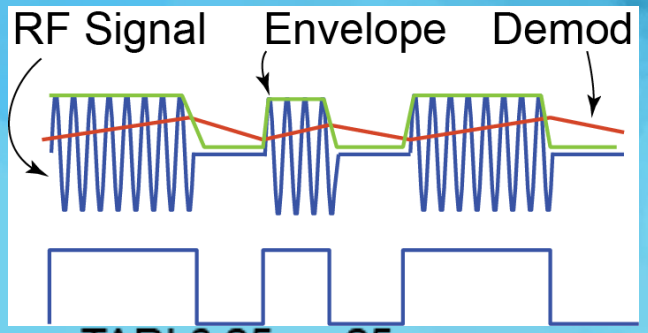


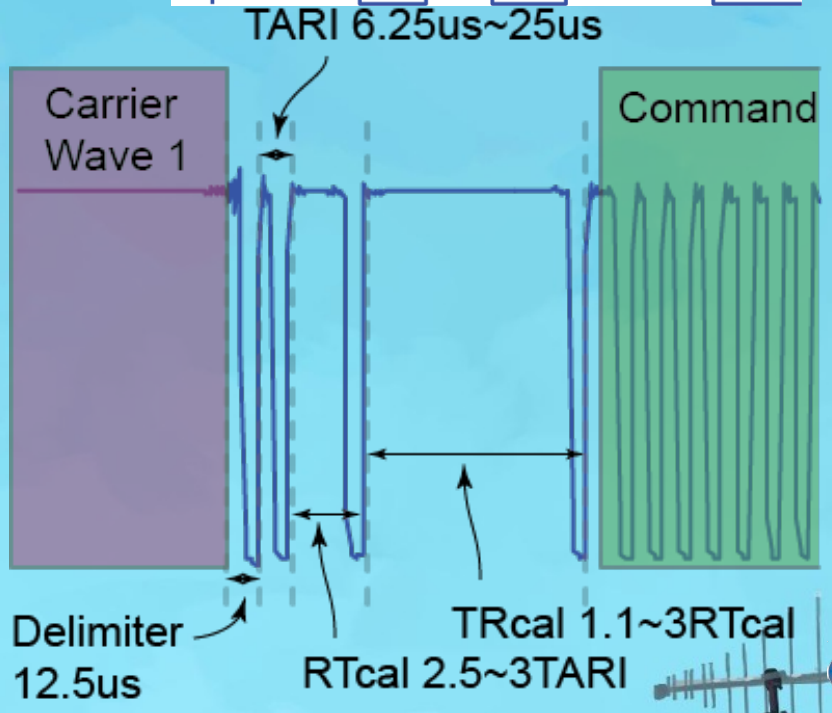


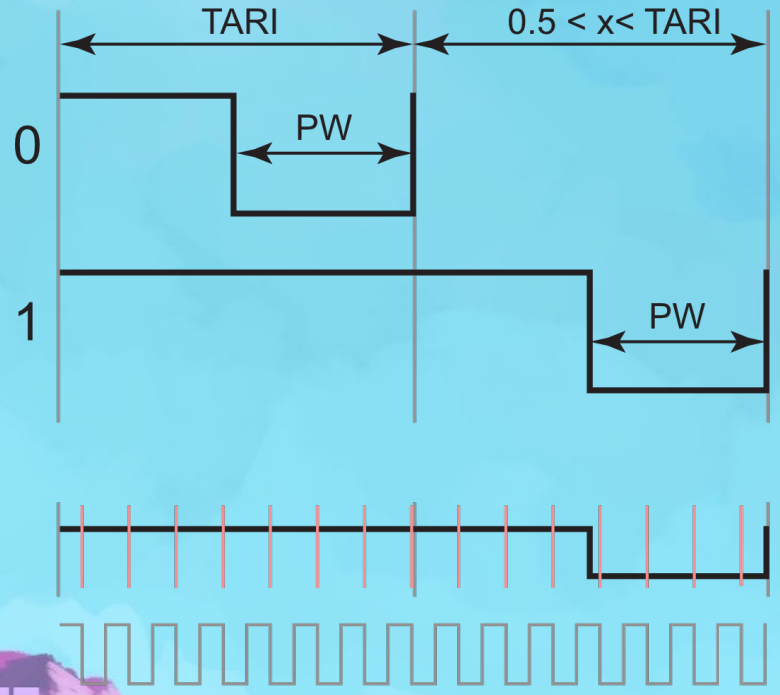
- This definitely works better with a resistor
- If you want to save money (area) diodes aren't a thing, you use MOSFETs.

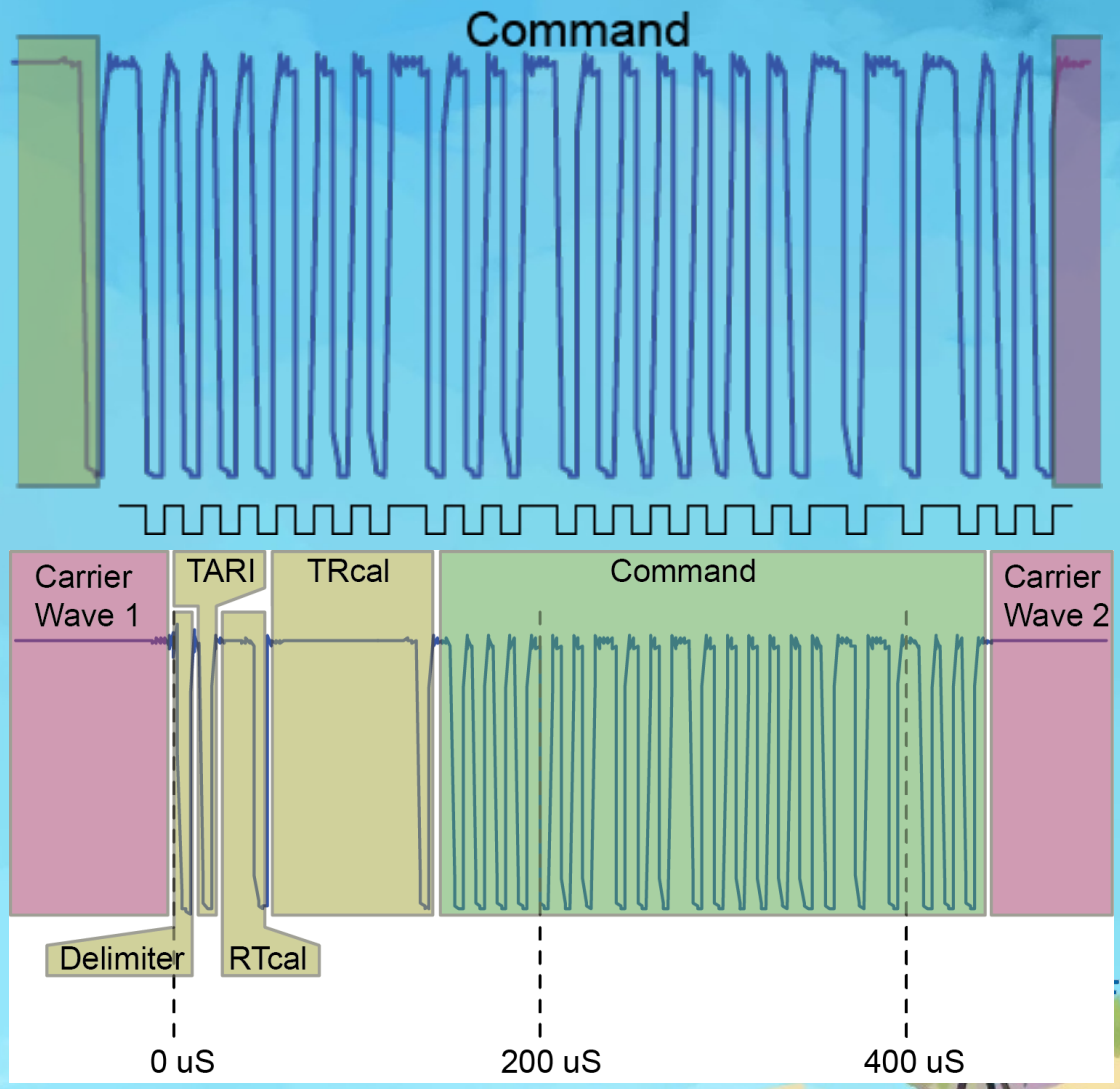
Data Extraction



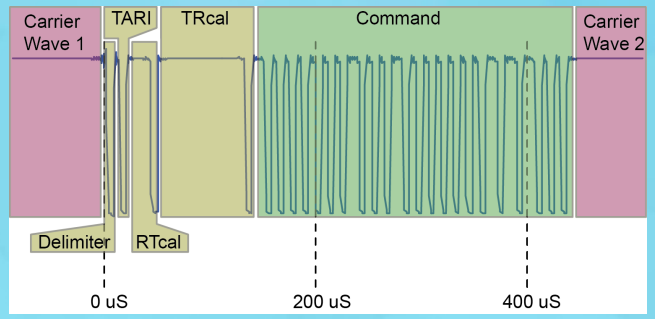


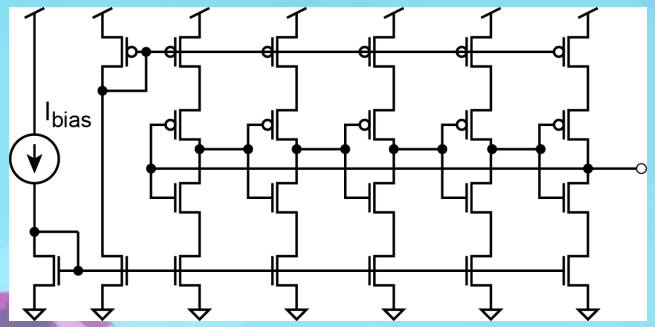


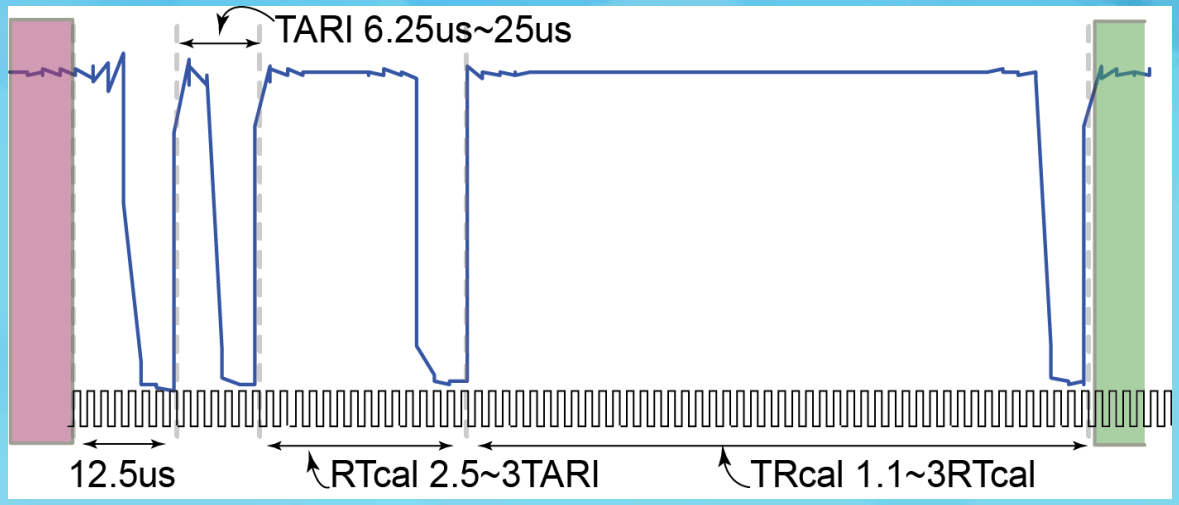

TARI
0.5 < x< TARI
0
PW
1
PW


Command
Carrier Wave 1
TARI
TRcal
Command
Carrier Wave 2
Delimiter
RTcal
0 uS
200 uS
400 uS

How many ticks per “bit”.







- Delimiter (Frame Start)
- Type A Reference Interval, “0”
- RTCAL: “0” + “1”
- TRCAL: Backscatter Link Frequency

Mandatory Inventory Commands

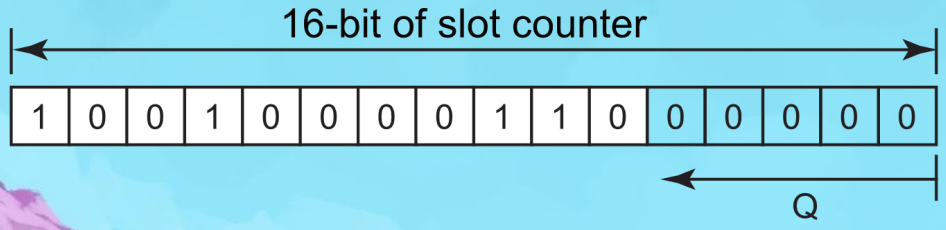
Command	Code	length (bits)	extra bits	Description
QueryRep	00	4	2-bit session	Advances the slot counter within a round
ACK	01	18	RNG/handle	AcknowledgesRN16 and requests EPC
Query	1000	22	payload	Starts an inventory round
QueryAdjust	1001	9	5-bit session/updown	Adjusts the Q parameter during a round
Select	1010	> 44	payload	Selects a tag population for subsequent inventory
NAK	11000000	8	none	Negative acknowledge

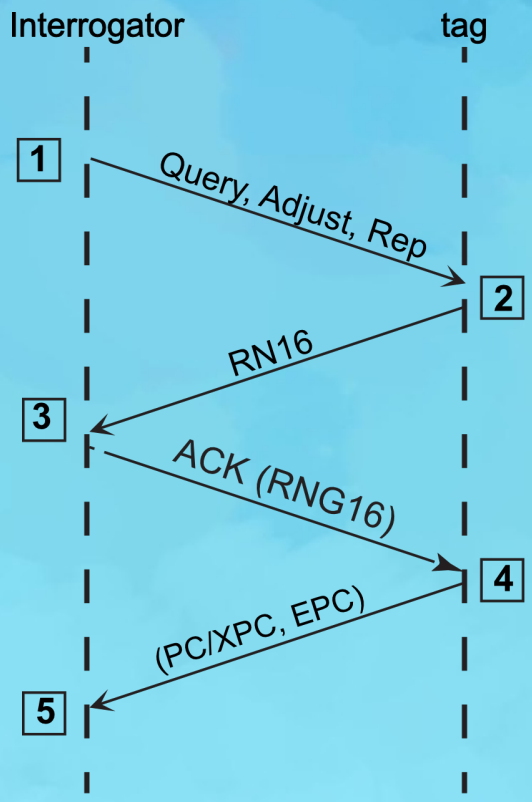
Mandatory Access Commands

Command	Opcode	Length (bits)	Extra Bits	Description
Req_RN	11000001	18	handle (RN16)	requests new RN16 from a singulated tag
Read	11000010	≥ 36	packet, CRC-16	read tag memory, 16-bit word(s)
Write	11000011	≥ 50	packet, CRC-16	write tag memory, one 16-bit word
Lock	11000100	52	packet, CRC-16	locks or unlocks memory and passwords
Kill	11000101	40	packet, CRC-16	permanently disables the tag
Access	11000110	40	packet, CRC-16	enters secured state for "protected operations"

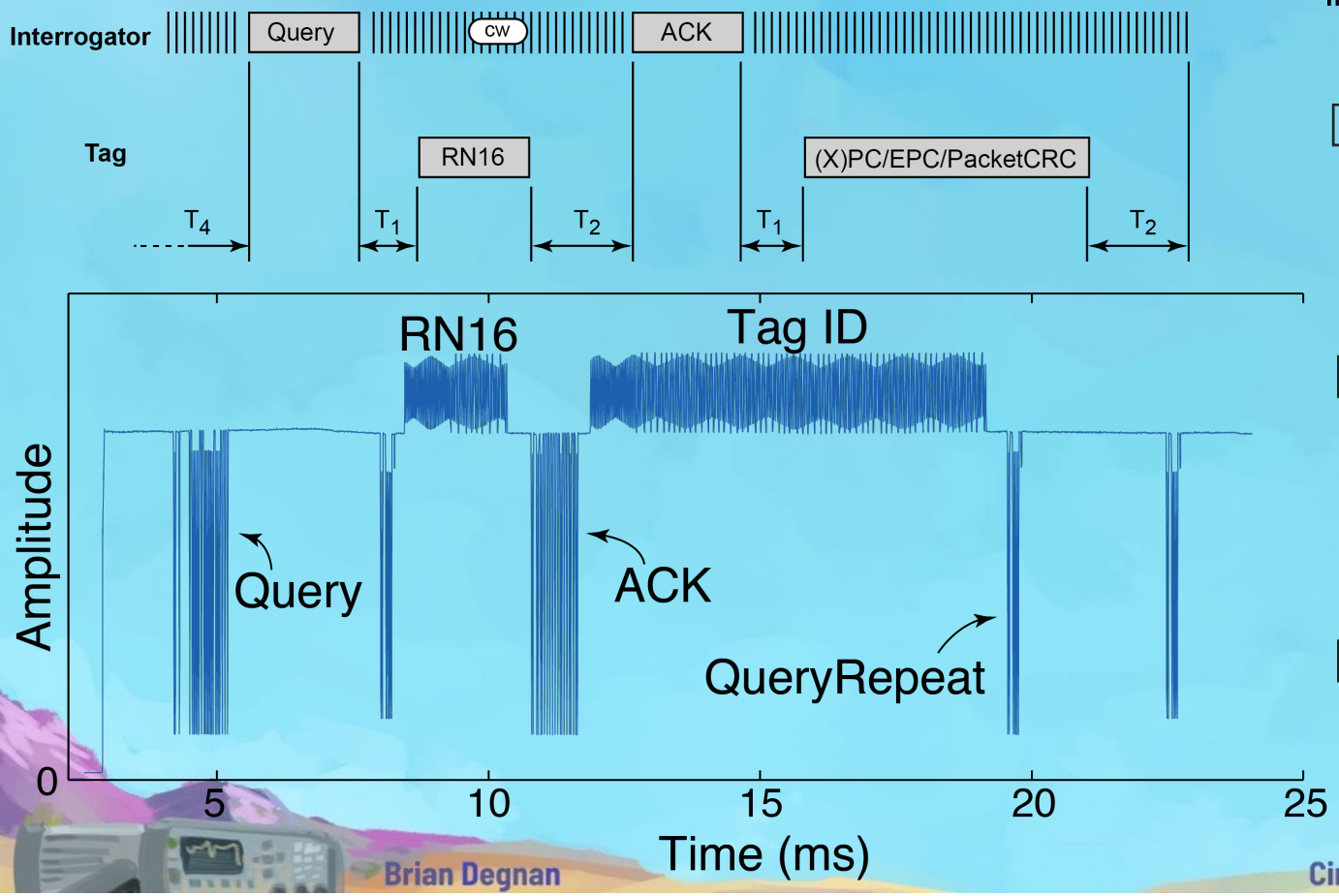
Time-Division Multiplexed System

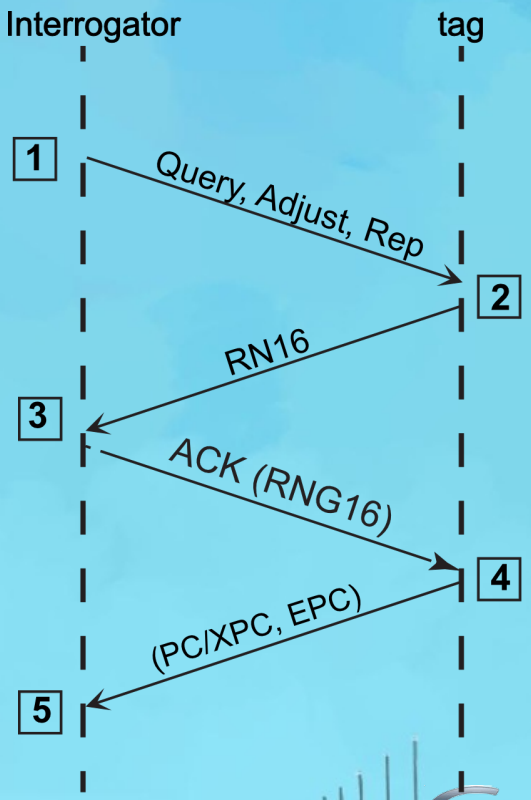
- Q-derived slot values
- Pseudo-Random Slot Selection
- Query initialized the "frame" by setting "Q" bit-depth (new RN16)
- QueryAdjust resizes the frame (new RN16)
- QueryRep advances through the frame





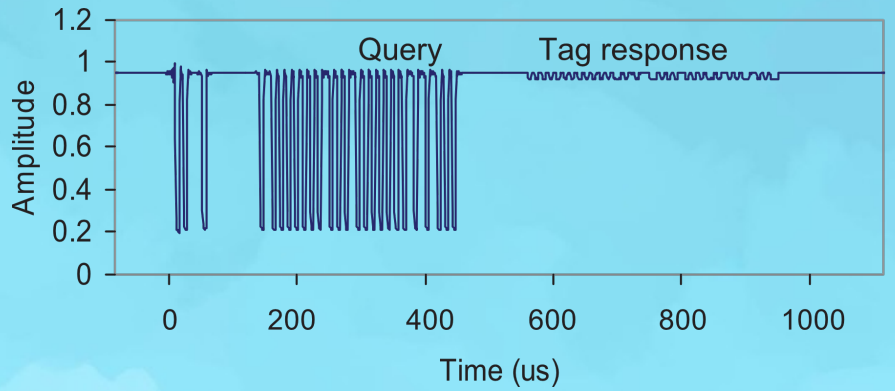
Commands in Context

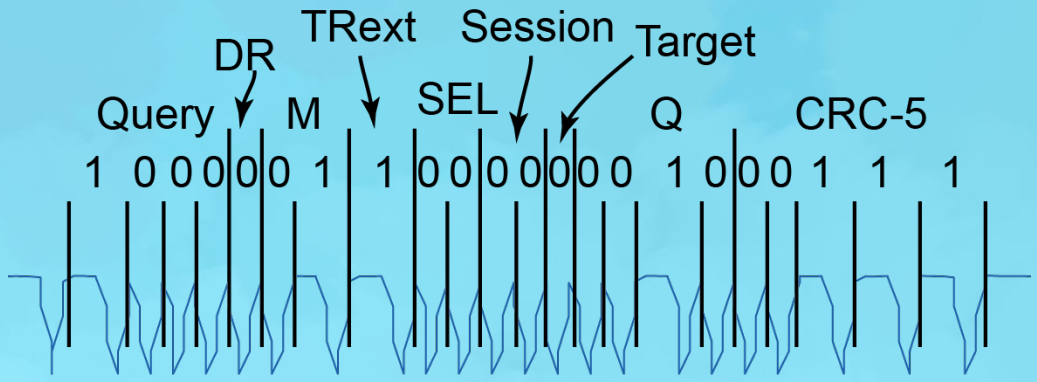




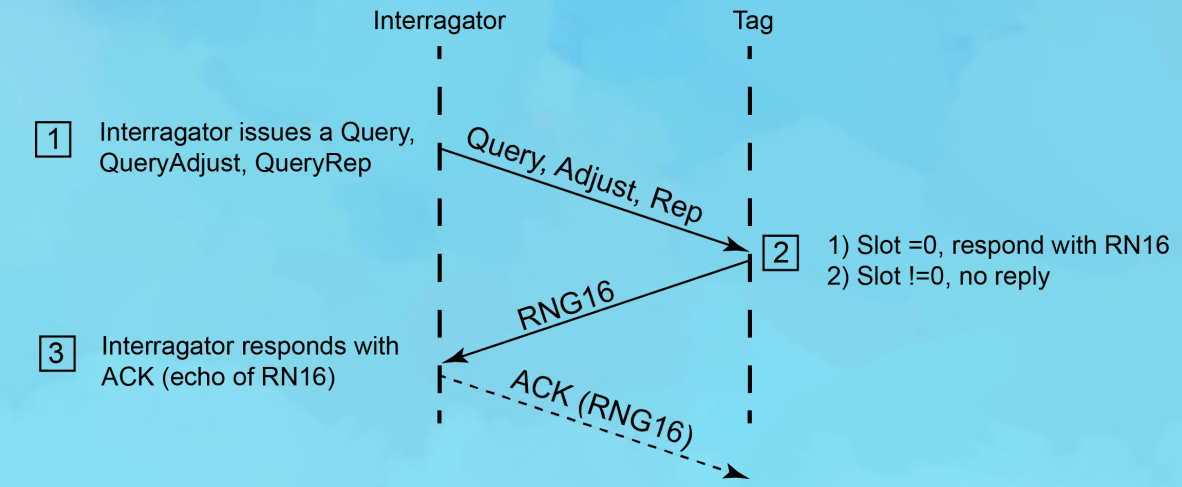
Query Bits

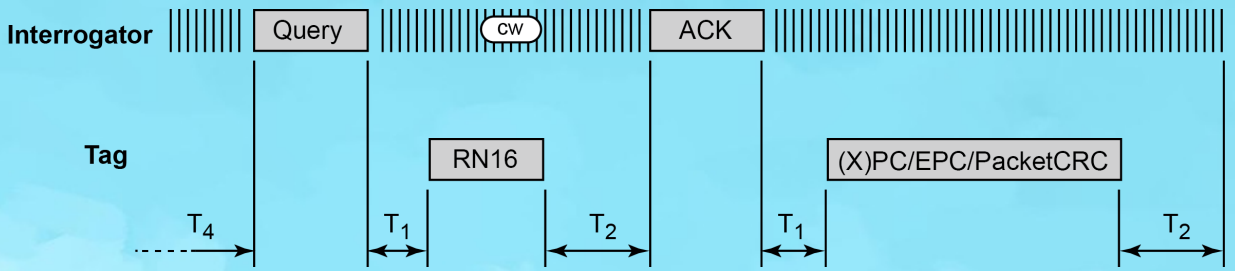
	Command	DR	M	TRext	Sel	Session	Target	Q	CRC
# of bits	4	1	2	1	2	2	1	4	5
description	1000	0: DR=8 1: DR=64/3	00: M=1 01: M=2 10: M=4 11: M=8	0: No pilot tone 1: Use pilot tone	00: All 01: All 10: ~**SL** 11: **SL**	00: S0 01: S1 10: S2 11: S3	0: *A* 1: *B*	0–15	CRC-5



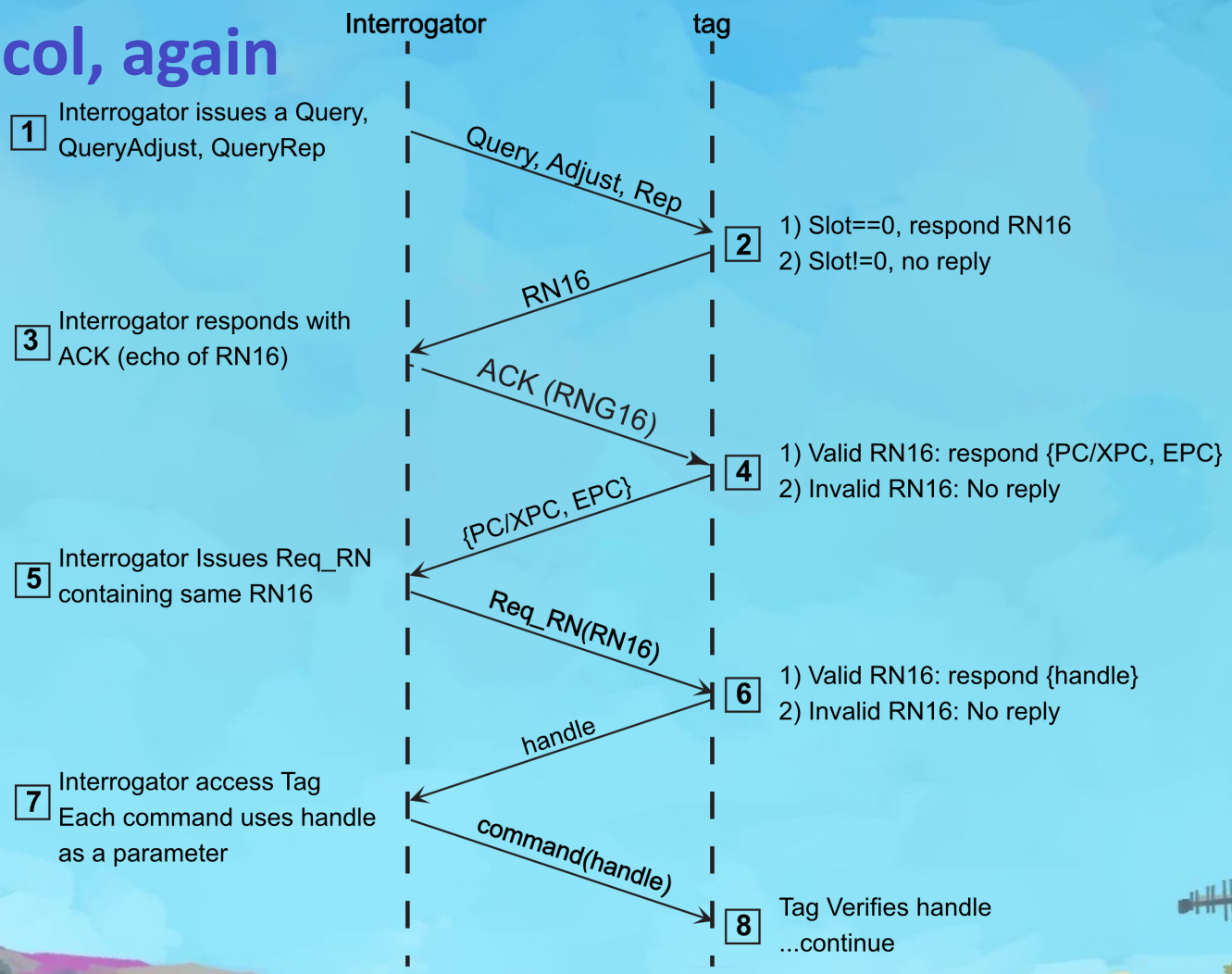


Gen2 State machine, getting the EPC.

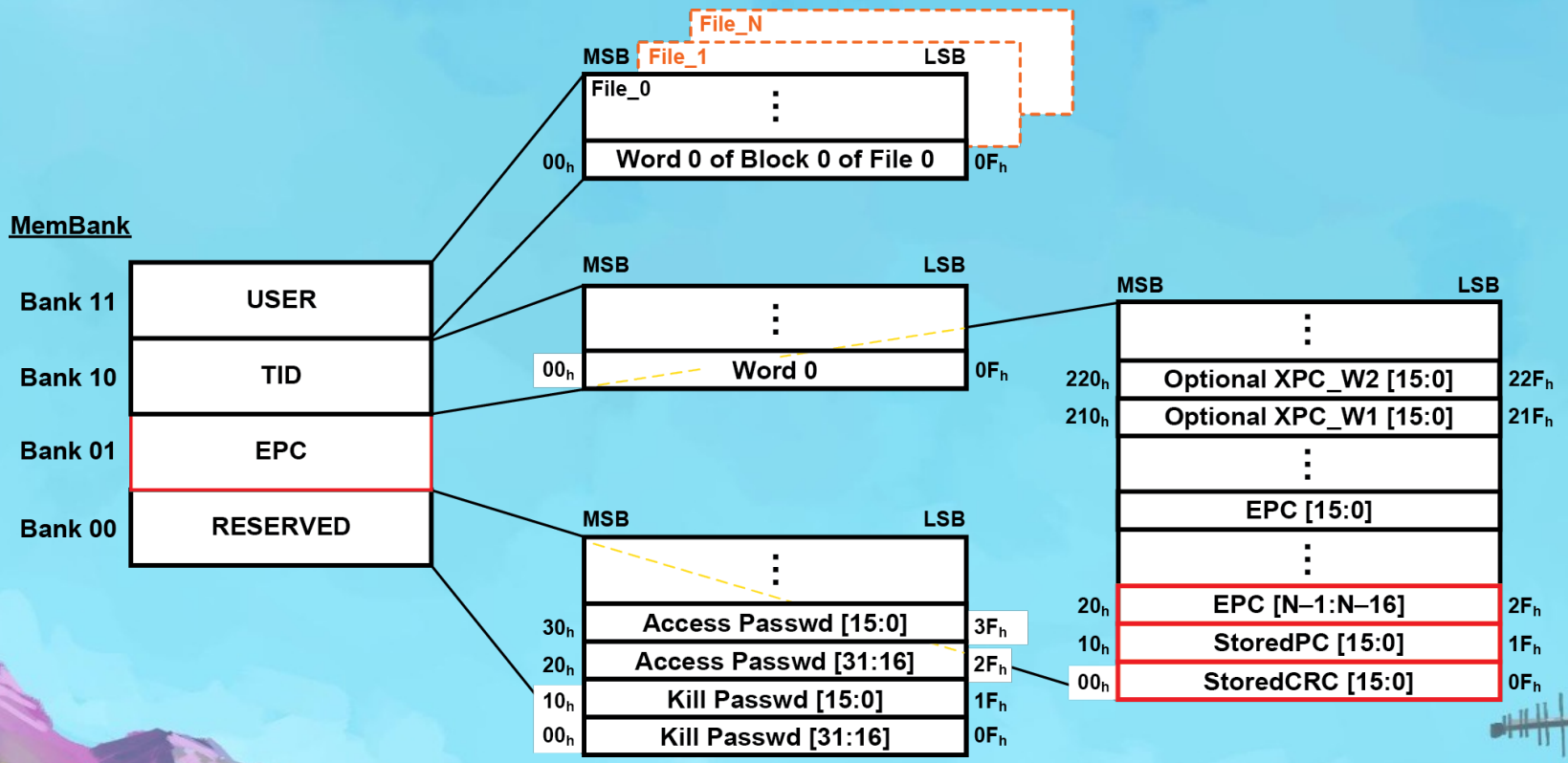




Gen2 Protocol, again



Tag Memory: What is stored?



EPC Memory

		EPC word values						
address	contents	0 words	1 words	2 words	3 words	4 words	5 words	6 words
0x00	StoredCRC	0xE2F0	0xCCAE	0x968F	0x78F6	0xC241	0x2A91	0x1835
0x10	StoredPC	0x0000	0x0800	0x1000	0x1800	0x2000	0x2800	0x3000
0x20	EPC word1	N/A	0x1111	0x1111	0x1111	0x1111	0x1111	0x1111
0x30	EPC word2	N/A	N/A	0x2222	0x2222	0x2222	0x2222	0x2222
0x20	EPC word3	N/A	N/A	N/A	0x3333	0x3333	0x3333	0x3333
0x40	EPC word4	N/A	N/A	N/A	N/A	0x4444	0x4444	0x4444
0x50	EPC word5	N/A	N/A	N/A	N/A	N/A	0x5555	0x5555
0x60	EPC word6	N/A	N/A	N/A	N/A	N/A	N/A	0x6666

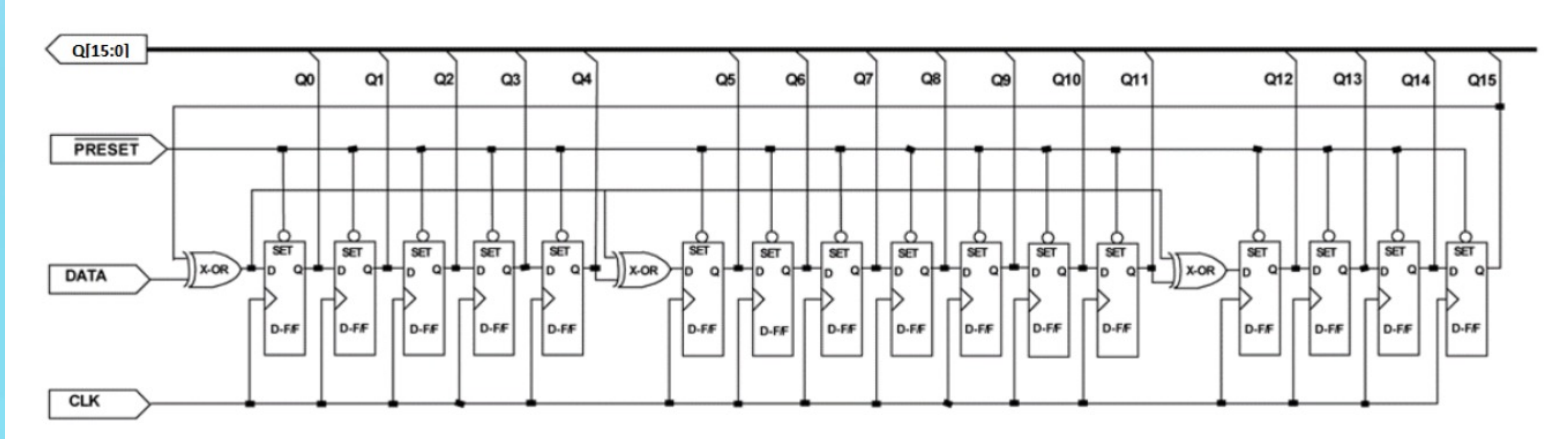
EPC Memory: StoredPC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L					UMI	XI	RFU / Flags								
EPC Length (as 16-bit words)					User Memory Indicator	Extended PC Indicator	Reserved for future use or protocol flags								

L (bits 15:11) — Number of 16-bit EPC words following the PC word (0–31).
UMI (bit 10) — Indicates presence of user memory (1 = user memory present).
XI (bit 9) — Indicates Extended PC words (XPC_W1/W2) are present.
Bits 8:0 — Reserved or protocol-defined flags depending on Gen2 version.

GF2, and CRCs

$$x^{16} + x^{12} + x^5 + 1$$



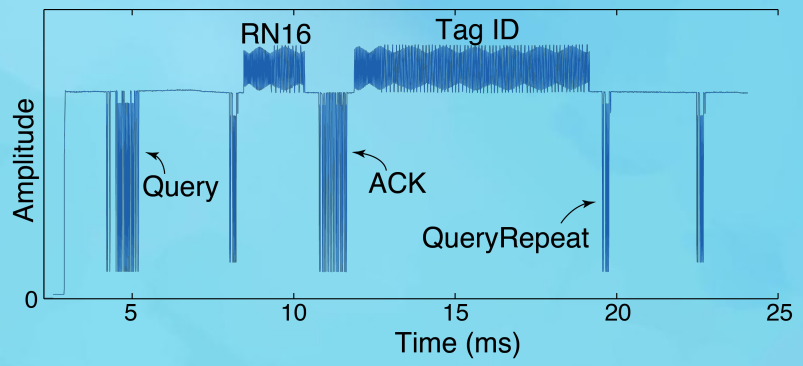
- Galois Field of 2 elements
- Addition is an XOR
- Multiplication is an AND
- Tiny in hardware!

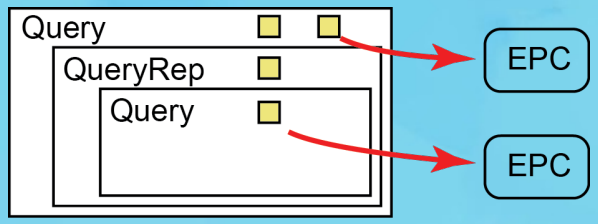
- Reset to 0xFFFF
- Shift in StoredPC
- Shift in the rest MSB first

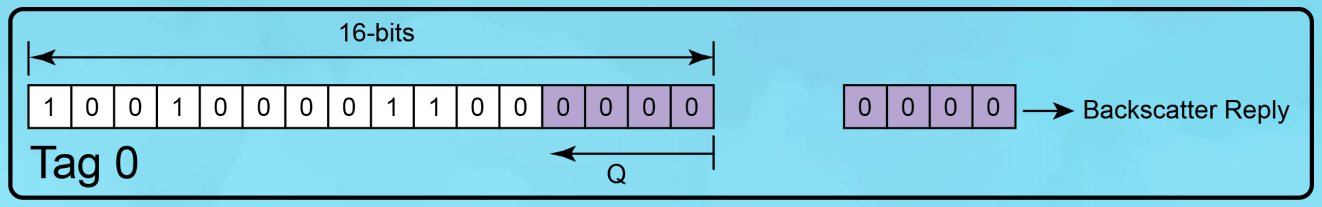
Complete inventory example

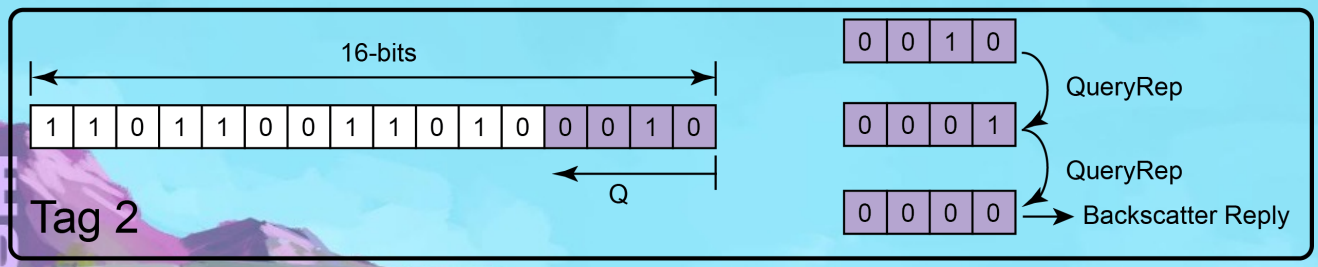
Message	Dir.	Fields (MSB-first)	Bitstream / Value
Query	R→Tag	Cmd=1000,DR=0, M=10, TRext=0, Sel=00, Session=00, Target=0, Q=0100 CRC-5	Data bits: 1000 0 10 0 00 00 0 0100 CRC-5: 10001 (0x11) Full: 10000100000000100 10001
RN16	Tag→R	RN16	RNG: 0x1234
ACK	R→Tag	Cmd=01,RN16	01 —— 0x1234
EPC Reply	Tag→R	PC —— EPC —— CRC-16	PC = 0x3000 (L=6 words⇒ 96-bitEPC) EPC = 0x1111 2222 3333 4444 5555 6666 CRC-16 over (PC——EPC) = 0x1835 Full reply: 3000 1111 2222 3333 4444 5555 6666 1835

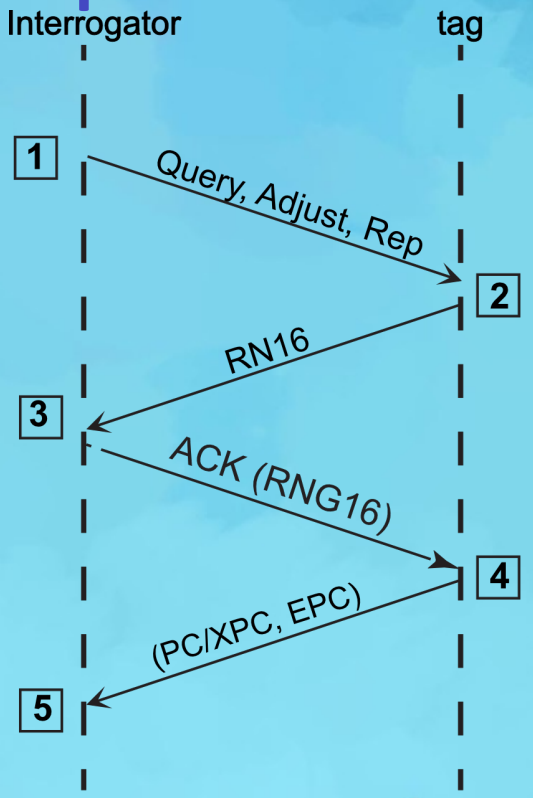
Slot Counter (the story of 2 tags) and QueryRep



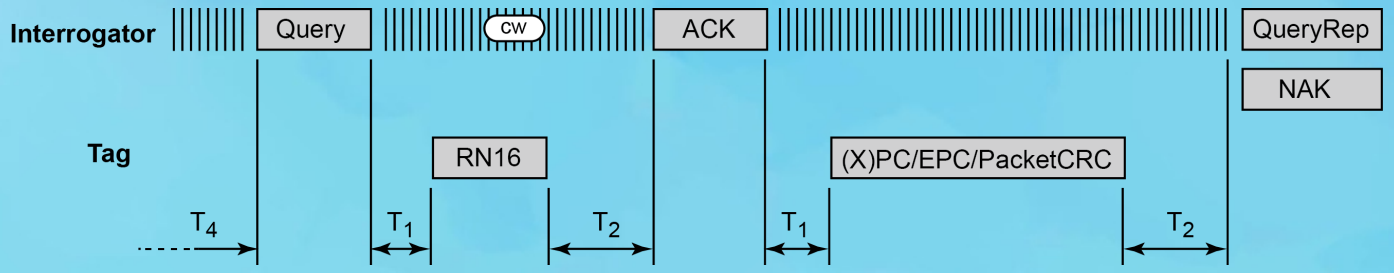


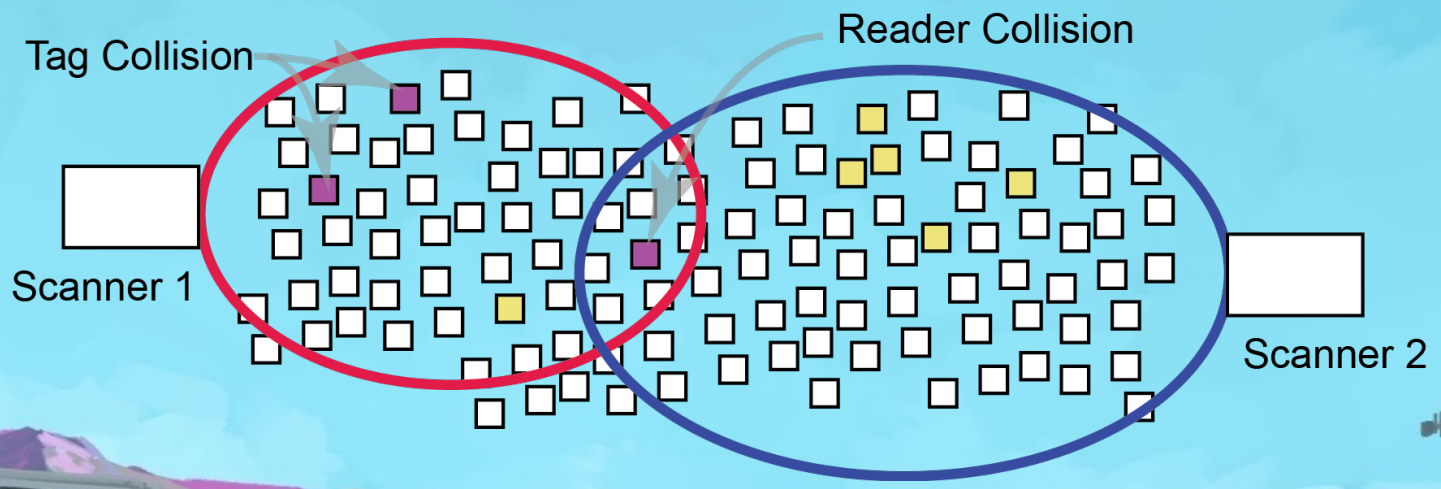




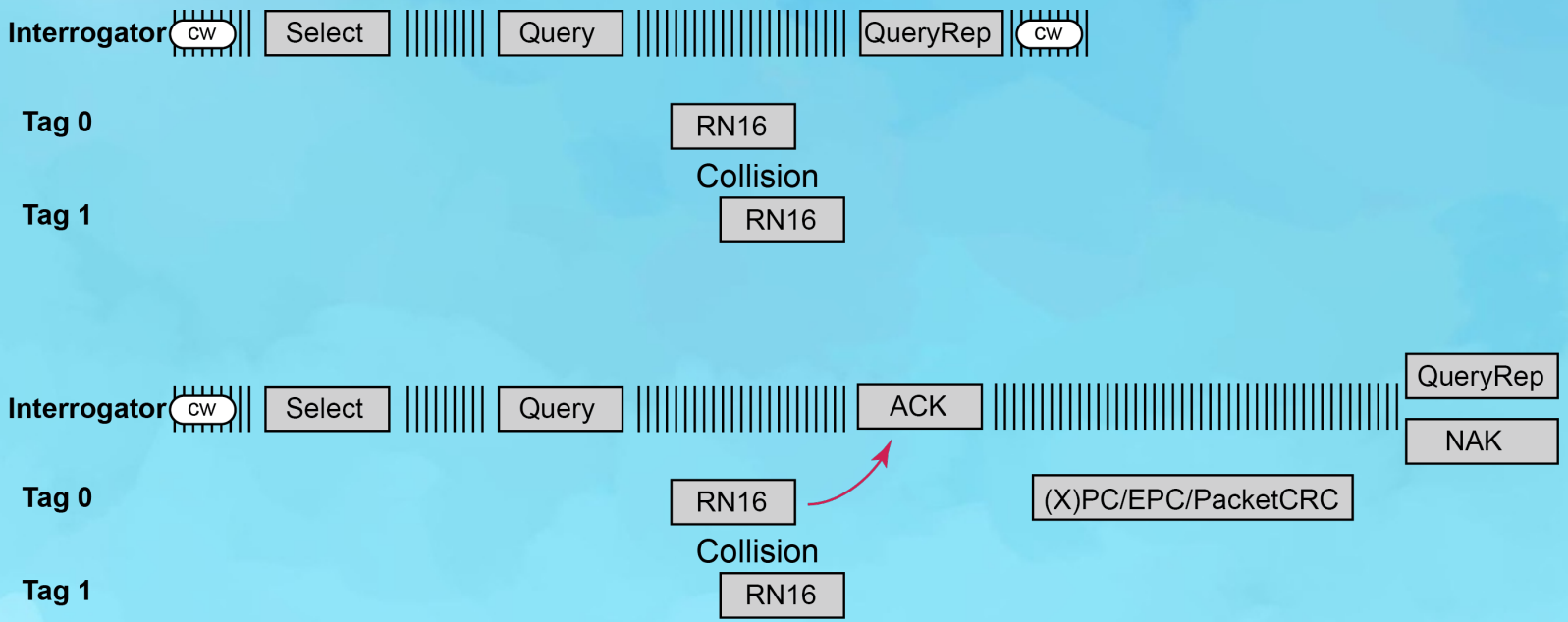


Collisions, they happen.

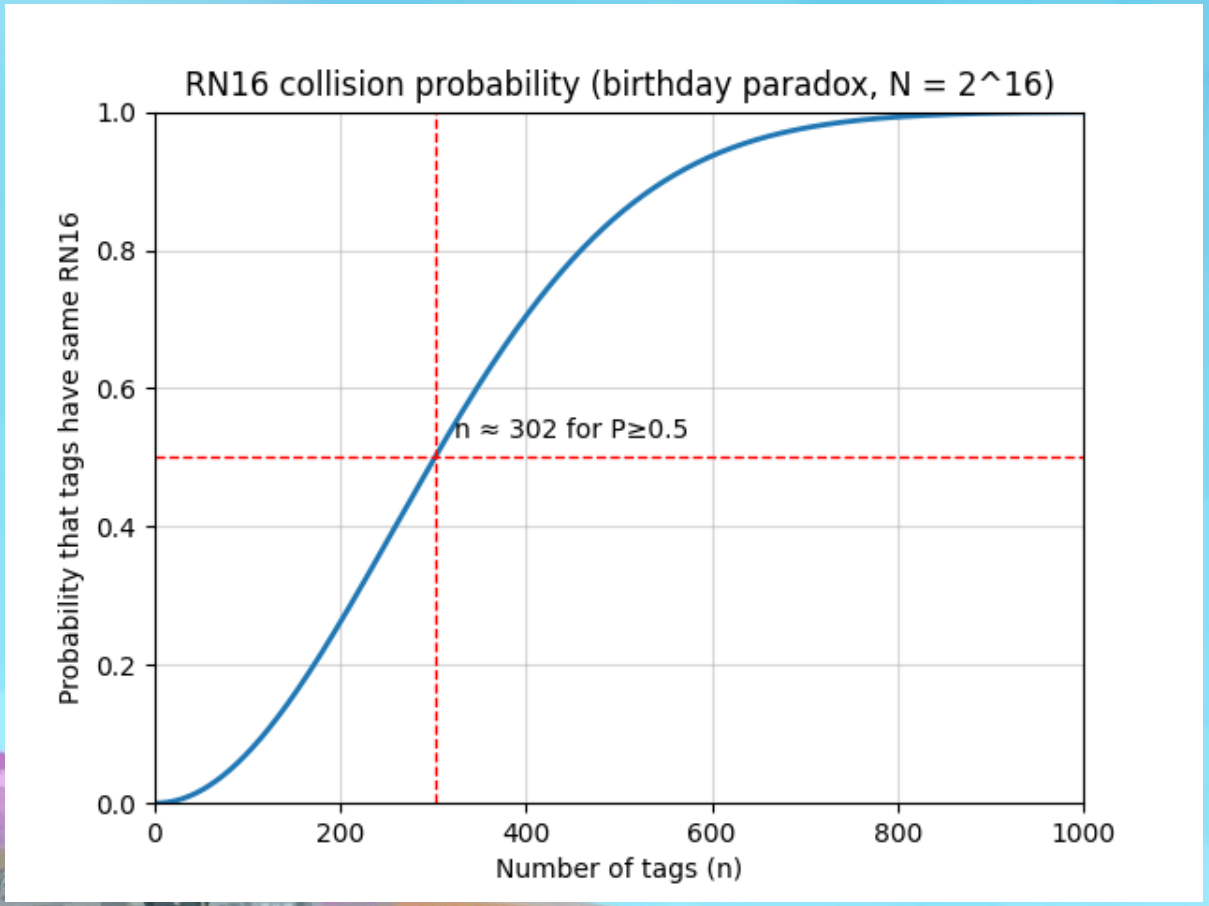




Packet Collision and Recovery



What are the chances?



$$p = 1 - e^{\frac{-k^2}{2N}}$$

$$p = 1 - e^{\frac{-k^2}{2(2^{16})}}$$

Circuits: Questions?

Monty Python's Life of Brian (1979)

Graph references for RFID semiconductor nodes

References

[1] U. Karthaus and M. Fischer, "Fully Integrated Passive UHF RFID Transponder IC with 16.7-μW Minimum RF Input Power," IEEE Journal of Solid-State Circuits, vol. 38, no. 10, pp. 1602–1608, Oct. 2003.

[2] T. Umeda et al., "A Passive UHF RF Identification CMOS Tag IC Using Ferroelectric RAM in 0.35-μm Technology," IEEE Journal of Solid-State Circuits, 2007.

[3] D. Drobny, G. Pailloncy, and N. Dehaese, "A Passive UHF RFID Transponder for EPC Gen2 With -14 dBm Sensitivity in 0.13-μm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, San Francisco, CA, USA, Feb. 2007.

[4] M. Baghaei-Nejad et al., "A Remote-Powered RFID Tag with 10 Mb/s UWB Uplink and -18.5 dBm Sensitivity UHF Downlink in 0.18-μm CMOS," IEEE ISSCC, 2008.

[5] Y. Hong, C. F. Chan, J. Guo, Y. S. Ng, W. Shi, L. K. Leung, K. N. Leung, C. S. Choy, and K. P. Pun, "Design of Passive UHF RFID Tag in 130 nm CMOS Technology," in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2008)*, Macao, China, Nov. 30–Dec. 3, 2008, pp. 1371–1374. :contentReference[oaicite:0]index=0

[6] J.-P. Curty, N. Joehl, C. Dehollain, and M. J. Declercq, "Remotely Powered Addressable UHF RFID Integrated System," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 11, pp. 2193–2202, Nov. 2005. :contentReference[oaicite:2]index=2

[7] W. H. Ki, C. Y. Tsui, and H. W. Leung, "RF Module Design of Passive UHF RFID Tag Implemented in CMOS 90-nm Technology," *Journal of Low Power Electronics*, vol. 6, no. 1, pp. 141–149, 2010. :contentReference[oaicite:1]index=1

[8] W. H. Ki et al., "RF Module Design of Passive UHF RFID Tag Implemented in CMOS 90-nm Technology," Journal of Low Power Electronics, vol. 6, no. 1, pp. 141–149, 2010.

[9] P. Han et al., "A 920-MHz Dual-Mode Receiver with Energy Harvesting for UHF RFID Applications," Electronics, vol. 9, no. 6, 2020.

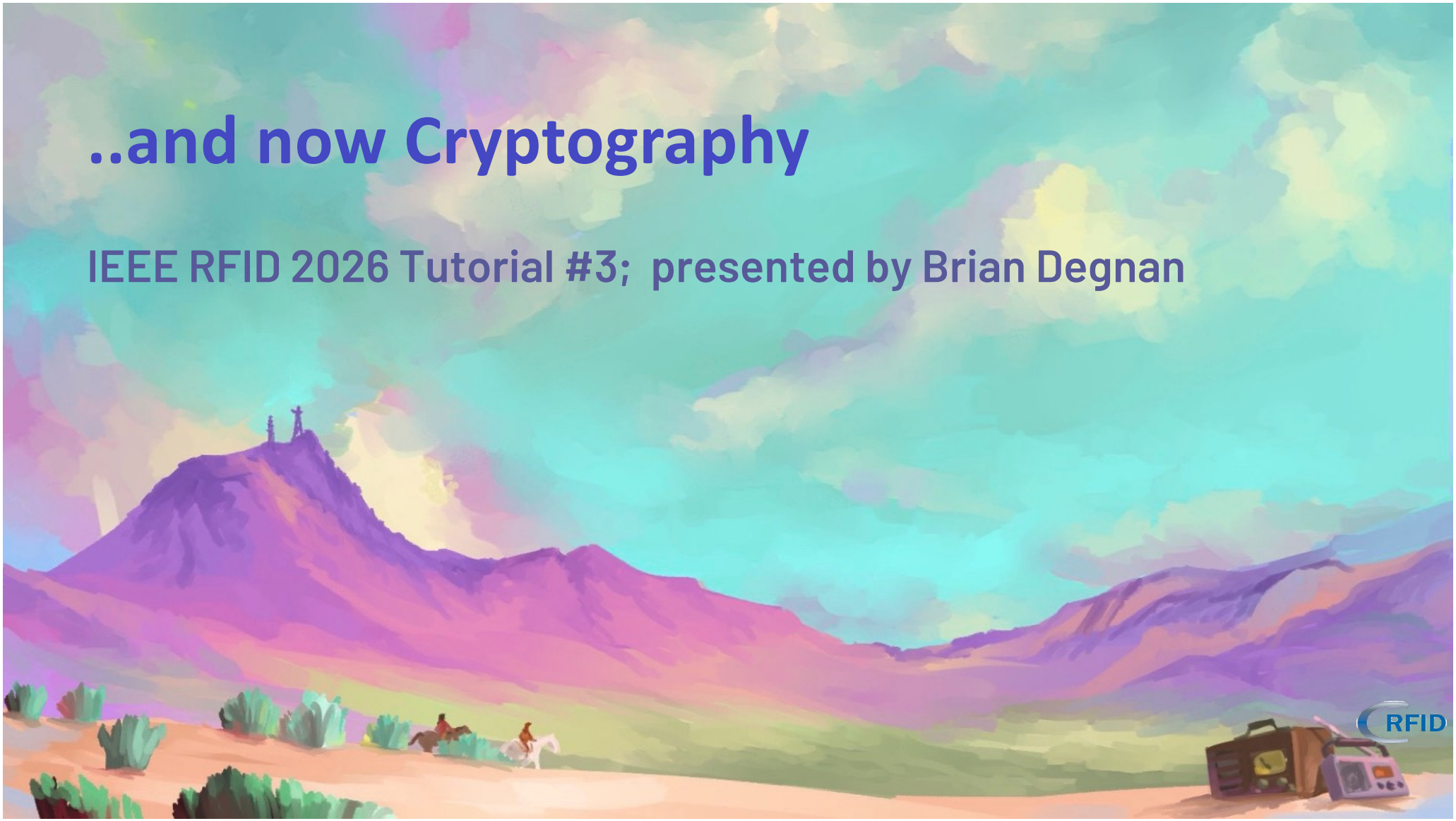
[10] S. Jinpeng et al., "Design and Implementation of an Ultra-Low Power Passive UHF RFID Tag," Journal of Semiconductors, vol. 33, no. 11, 2012.

[11] NXP Semiconductors, "UCODE RAIN RFID Product Family Datasheets."

[12] Impinj, Inc., "Monza R6 Series RAIN RFID Endpoint ICs."

..and now Cryptography

IEEE RFID 2026 Tutorial #3; presented by Brian Degnan

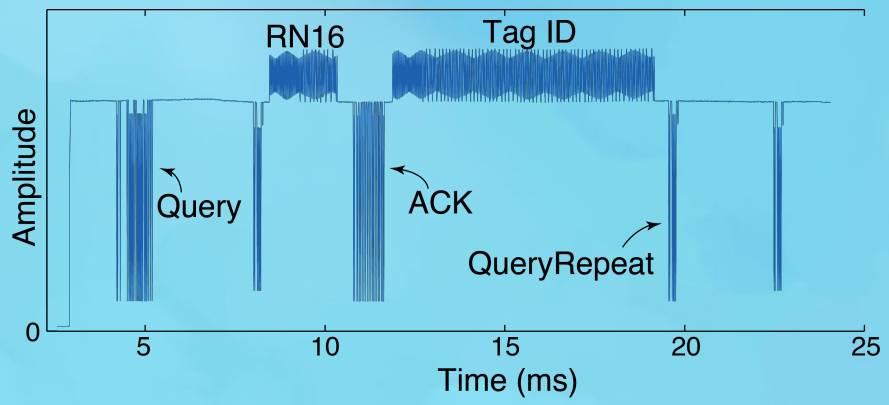


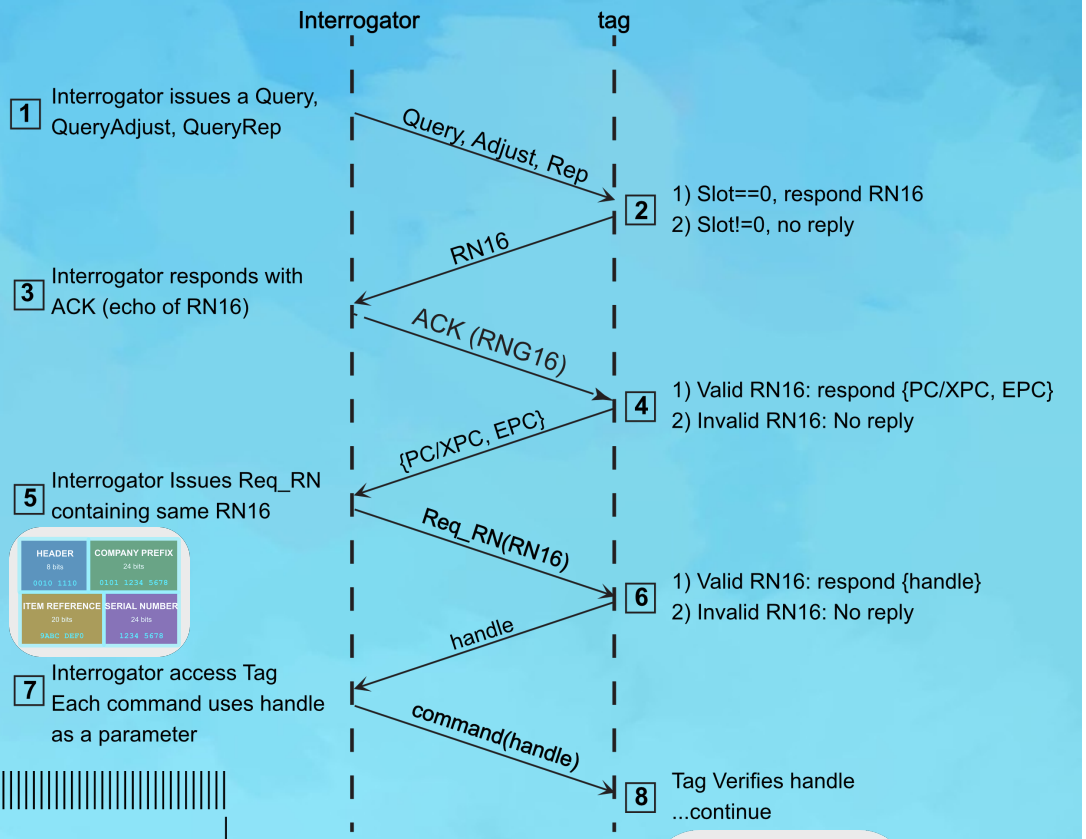
Cryptography! What is it?

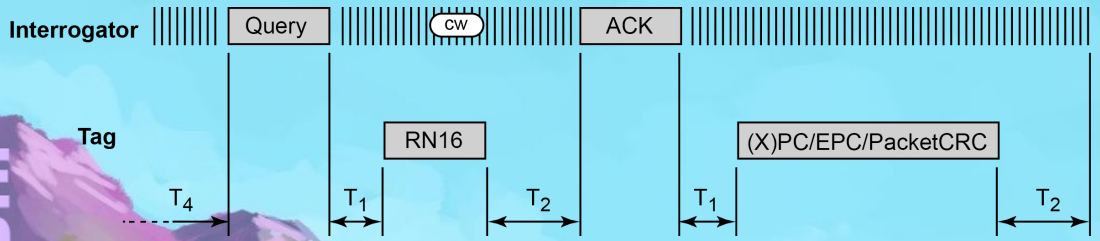
- Cryptography: Making data look random through a “secret”.
- Cryptanalysis: Getting the data without the “secret”.

- Is it math? Is it policy? It is magic?
- Crypto: Whatever the internet decides it is that day…

Gen2 Protocol, again







Digital Dreams?

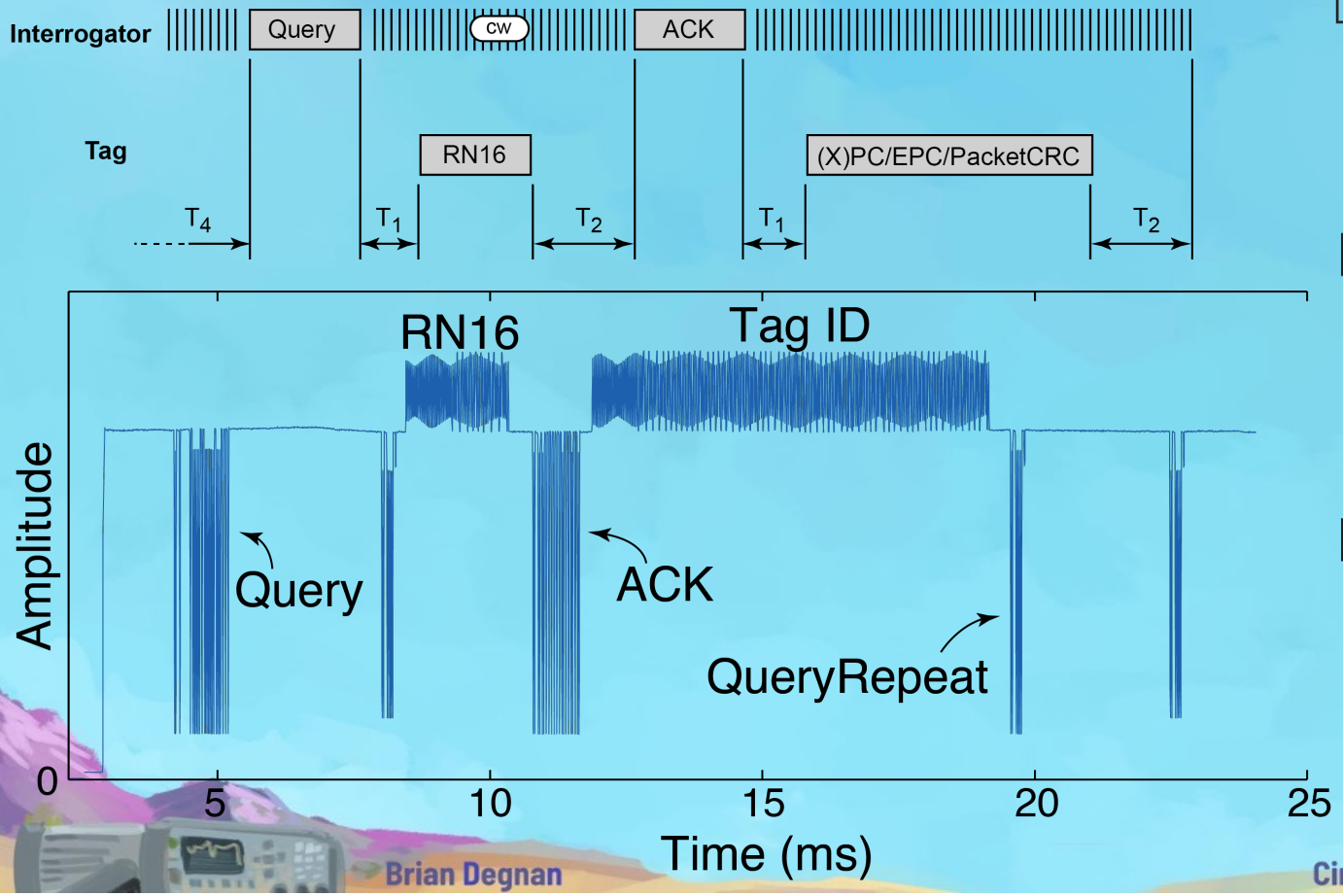
https://www.gs1.org/standards/rfid/uhf-air-interface-protocol

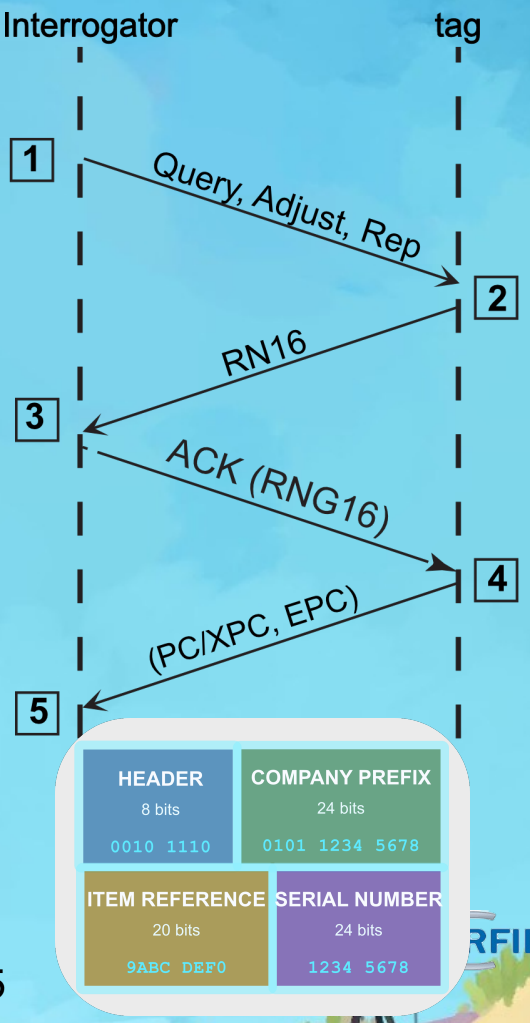
Cryptography in Gen2V2

- Gen2v2 has NONE (by design)
- User's choice of underlying cryptography
- Defines the commands and responses

Command	Code	Description
Authenticate	11010101	Authenticate against Challenge
AuthComm	11010111	Encapsulate command securely
Challenge	11010100	Crypto "Select" command
KeyUpdate	11000101	Update an encryption key
SecureComm	11010110	Encapsulate command securely
TagPrivilege	1110001000000011	Set tag access
Untraceable	1110001000000000	Hide from other interrogators

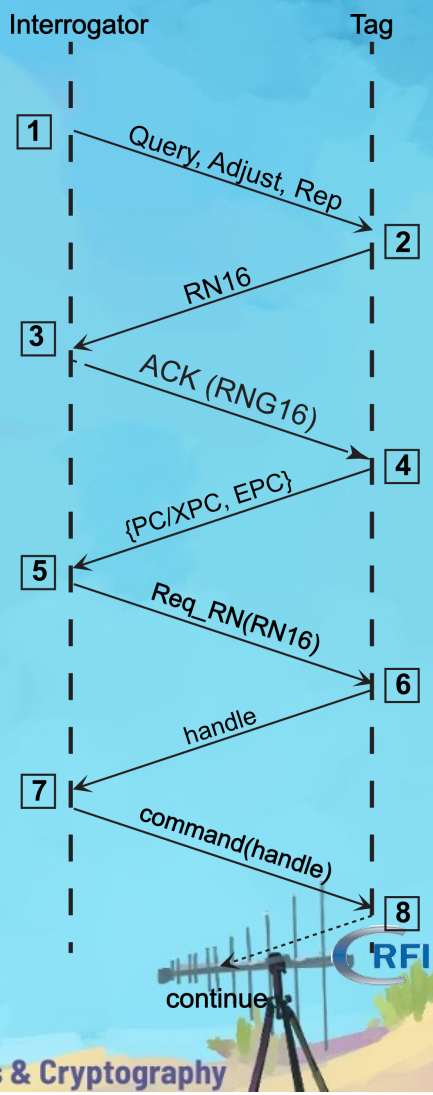
Reminder of Standard Enumeration



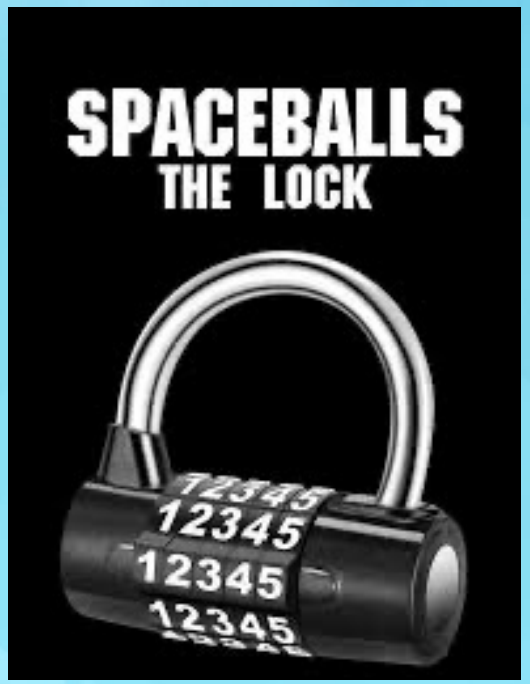


Complete Access Example

Message	Dir.	Fields	Value (MSB-first)
Query	R→Tag	Cmd=1000,DR=0, M=10, TRext=0, Sel=00, Session=00, Target=0, Q=0100 CRC-5	Data bits: 1000 0 10 0 00 00 0 0100 CRC-5: 10001 (0x11) Full: 1000010000000000100 10001
RN16	Tag→R	Random number	0x1234
ACK	R→Tag	Cmd=01,RN16	01 1234
EPC Reply	Tag→R	PC —— EPC —— CRC-16	PC = 0x3000(L=6) EPC = 1111 2222 3333 4444 5555 6666 Reply CRC-16 = 0x1835 Full: 3000 1111 2222 3333 4444 5555 6666 1835
Req_RN	R→Tag	Opcode=11000001,Handle,CRC-16	C1 1234 <CRC>
New RN16	Tag→R	New handle	0xABCD
ACCESS	R→Tag	Opcode=11000010,CoveredPwd[31:16], CoveredPwd[15:0], Handle, CRC-16	Password = 1234 5678 Handle = ABCD Covered1 = 1234 XOR ABCD = B9F9 Covered2 = 5678 XOR ABCD = FDB5 CRC-16 = 0x3A5E Full: C2 B9F9 FDB5 ABCD 3A5E



You XOR the password with the handle….



ROT13 of “FCNPRONYYF GUR YBPX” is

SPACEBALLS THE LOCK

Of course:

Znl gur Fpujnegm or jvgu lbh.

The problem with “Cover Codes”.

$$P_1 = 0x1234$$
$$P_2 = 0x5678$$
$$K = 0xABCD$$

$$C_1 = P_1 \oplus K = 0xB9F9$$
$$C_2 = P_2 \oplus K = 0xFDB5$$

$$C_1 \oplus C_2 = 0xB9F9 \oplus 0xFDB5 = 0x444C$$
$$P_1 \oplus P_2 = 0x1234 \oplus 0x5678 = 0x444C$$

Introduction to “actual” Cryptography

- Asymmetric
- Public/private key
- Send over insecure channel
- Bit widths 160~4096
- Relatively Slow
- Many ciphers but RSA/ECC/Diffie are common

- Symmetric
- Private Key
- Uses Asymmetric cryptography to send key
- Bit widths 64~256
- Quite Fast
- Many ciphers but AES is the standard

Simple Asymmetric Example (HTTPS uses this type)

Me	You
$Prime, P = 29, \alpha = 2$	
$secret_{me} = 5$	$secret_{you} = 12$
$\alpha^{secret_{me}} = 2^5 = 32$	$\alpha^{secret_{you}} = 2^{12} = 4096$
$= x \pmod P$	$= y \pmod P$
$= 3 \pmod{29}$	$= 7 \pmod{29}$
$y^{secret_{me}} = Z \pmod P$	$x^{secret_{you}} = Z \pmod P$
$7^5 = Z \pmod{29}$	$3^{12} = Z \pmod{29}$
$Z = 16$	

Encryption Example of AES128 (0x10 vs. 0x11)

Key (Z of 16) = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 10

Plaintext = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Ciphertext = c6 a1 3b 37 87 8f 5b 82 6f 4f 81 62 a1 c8 d8 79

Key (Z of 17) = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 11

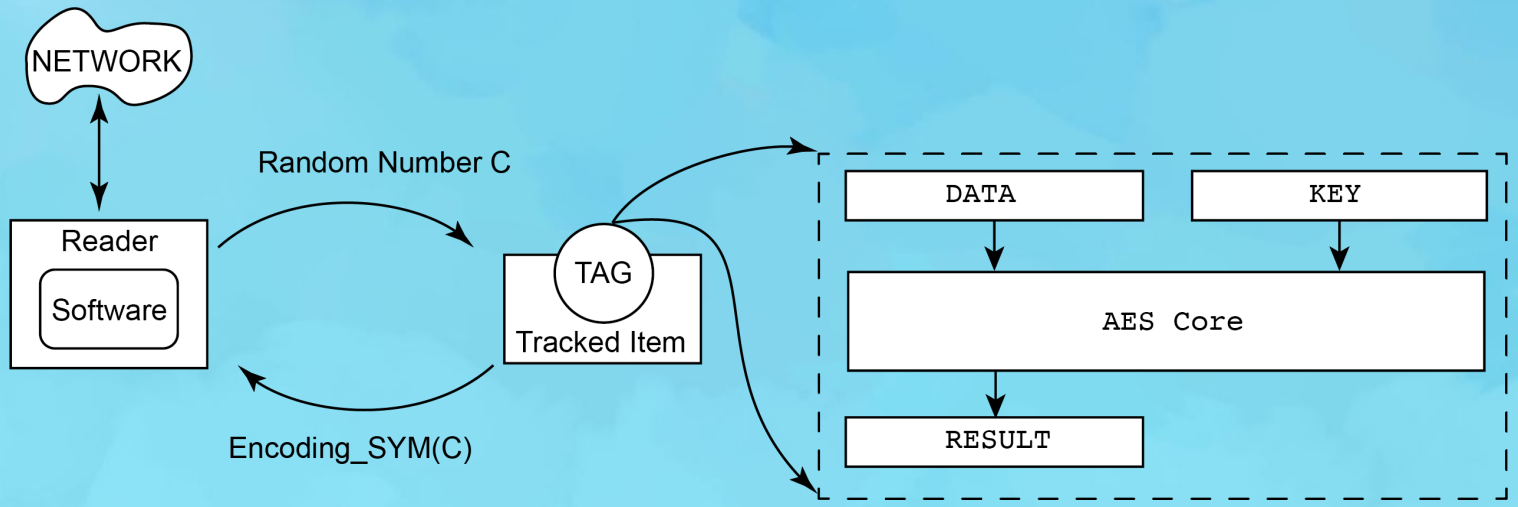
Plaintext = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Ciphertext = d0 5f 7c 3b 5a 9d 2e 89 6c 7a b8 1e 2f 4c 92 a3

Cryptography in Gen2V3…same as Gen2V2

- Memory better defined
  - Out of range addresses, remaining user memory
  - XPC ambiguity resolved
- RFU bits are not to be used
  - Some tags only worked with certain readers
  - PCWord RFU bits were being used as capability markers
- CSI (Cryptographic Suite Indicator) rules are tightened for hierarchy.

How crypto tags work… Authentication example



C=00000000000000000000000000000000

Key=00000000000000000000000000000000

Enc=66e94bd4ef8a2c3b884cfa59ca342b2e

Message	Dir.	Fields (MSB-first)	Bitstream / Value
Query	R→Tag	Cmd=1000,DR=0, M=10, TRext=0, Sel=00, Session=00, Target=0, Q=0100 CRC-5	Data bits: 1000 0 10 0 00 00 0 0100 CRC-5: 10001 (0x11) Full: 1000010000000100 10001
RN16	Tag→R	RN16	RNG: 0x1234
ACK	R→Tag	Cmd=01,RN16	01 ‖0x1234
EPC Reply	Tag→R	PC ‖EPC ‖CRC-16	PC = 0x3000 (L=6 words⇒ 96-bitEPC) EPC = 0x1111 2222 3333 4444 5555 6666 CRC-16 over (PC‖EPC) = 0x1835 Full reply: 3000 1111 2222 3333 4444 5555 6666 1835
Req_RN	R→Tag	Cmd=11000001,RN16, CRC-16	RN16 from inventory: 0x1234 CRC-16 Full: C1 1234 CRC16
Handle Reply	Tag→R	Handle (new RN16)	0x5678
Authenticate	R→Tag	Cmd=11010101,RFU=00, SenRep=1, IncRepLen=0, CSI=0x01, Length=0x080 (128 bits, 12-bit field), Message=128-bit challenge, Handle=0x5678, CRC-16	Challenge: 00000000000000000000000000000000 Handle: 0x5678 CRC-16 (over all fields) Note: Authenticate uses an in-processreply (CW follows).
Auth. Reply	Tag→R	Barker ‖Done‖ Header ‖Response ‖ CRC-16	Barker: 1110010 Done=1 (finished), Header=0 (success) Response (example suite): 66e94bd4ef8a2c3b884cfa59ca342b2e Handle: 0x5678 CRC-16 over (Done ‖Header ‖Response ‖Handle)

Seems simple, but we need to talk about time.

- Barker Code of b1110010 (hang with me)
- Tradeoff: balance throughput/power/area

Message	Dir.	Fields (MSB-first)	Bitstream / Value
...	...	...	
Handle Reply	Tag→R	Handle (new RN16)	0x5678
Authenticate	R→Tag	Cmd=11010101,RFU=00, SenRep=1, IncRepLen=0, CSI=0x01, Length=0x080 (128 bits, 12-bit field), Message=128-bit challenge, Handle=0x5678, CRC-16	Challenge: 00000000000000000000000000000000 Handle: 0x5678 CRC-16 (over all fields) Note: Authenticate uses an in-process reply (CW follows).
Auth. Reply	Tag→R	Barker ||Done|| Header ||Response || CRC-16	Barker: 1110010 Done=1 (finished), Header=0 (success) Response (example suite): 66e94bd4ef8a2c3b884cfa59ca342b2e Handle: 0x5678 CRC-16 over (Done ||Header ||Response ||Handle)

Message	Dir.	Fields (MSB-first)	Bitstream / Value
…	…	…	
Handle Reply	Tag→R	Handle (newRN16)	0x5678
Authenticate	R→Tag	Cmd=11010101, RFU=00, SenRep=1, IncRepLen=0, CSI=0x01, Length=0x080 (128 bits, 12-bit field), Message=128-bit challenge, Handle=0x5678, CRC-16	Challenge: 00000000000000000000000000000000 Handle: 0x5678 CRC-16 (over all fields)
Barker Reply 1	Tag→R	Barker || Done || Header	Time: $t_1 \leq T6_{max}$ (20ms) Barker: 1110010 Done=0 (still computing) Header=0
Barker Reply 2	Tag→R	Barker || Done || Header	Time: $t_2 - t_1 \leq T7_{max}$ (20ms) Barker: 1110010 Done=0 (still computing) Header=0
Authenticate Reply (Final)	Tag→R	Barker || Done || Header || Response || Handle || CRC-16	Time: $t_3 - t_2 \leq T7_{max}$ (20ms) Barker: 1110010 Done=1 (finished) Header=0 (success) Response: 66e94bd4ef8a2c3b884cfa59ca342b2e Handle: 0x5678 CRC-16 over (Done || Header || Response || Handle)

What are the choices for cryptography? ISO/IEC 29167

RFID CSI	ISO/IEC Part	Name	Status
0	Reserved	Reserved	RFU
1	29167-10	AES-128	Published / Active
2	29167-11	PRESENT-80	Published / Active
3	29167-12	ECC-based Authentication	Published / Active (Asymmetric)
4	29167-13	Grain-128A	Published / Active
5	29167-14	CryptoGPS	Published / Active (Asymmetric)
6	29167-15	XTEA	Published / Active
7	29167-16	ECDSA / ECSchnorr	Published / Active (Asymmetric)
8	29167-17	SHA-based Challenge-Response	Published / Active (Hash-based)
9	29167-18	RAMON	Published / Active
10	29167-19	KATAN	Published / Active
11	29167-20	Simon	Published / Active
13	29167-22	Keccak	Published / Active
14	29167-23	ASCON	NIST 4118
15	Reserved	Reserved	RFU

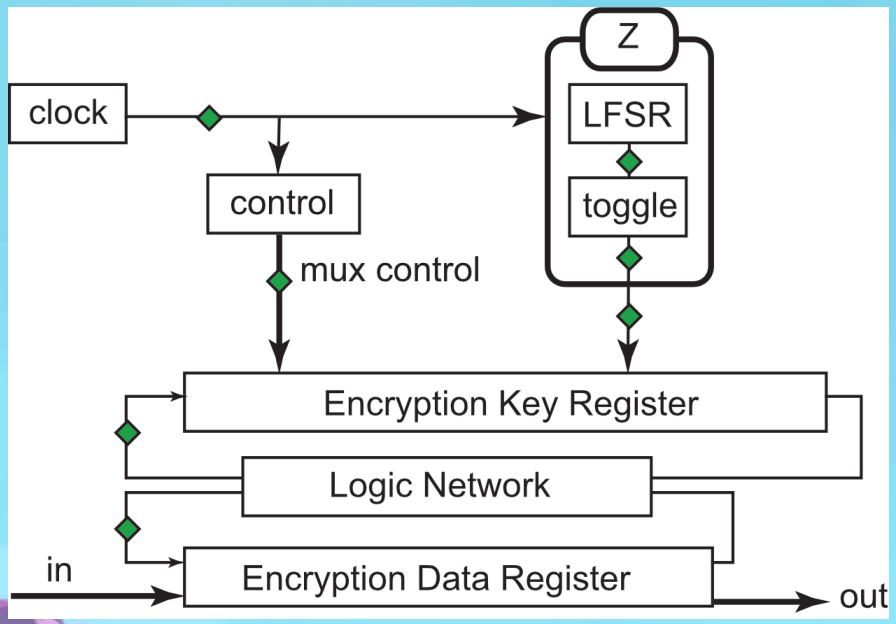
The cryptography problems?

- Passive RFID is very power constrained
  --need power appropriate ciphers (PRESET/SIMON for example)
- Cryptographic Random numbers are a challenge.
- Active RFID has more options:
  --AES… and everything else can be used
- Different use-cases prioritize different performance attributes
- Different use-cases have different security/threat models
- Deployment is complex
- Symmetric cipher: lowest complexity, hard key exchange policy

Cryptography on Passive RFID (it's a $10 t-shirt)

- No device authentication (who's talking?)
- Practical security depends more on implementation and cost than nominal key length.
- Security cost must match product economics (ex: disposable retail tags).
- Tags often only used a few times
- cryptography primarily establishes tag authenticity rather than data confidentiality.

One option:
SIMON128/128: A “simple” block cipher



- Clock it off carrier?
- Make is as small as possible?
- Minimize the bit requirements?

Structure of a block cipher is byte blocks (key example)

block size (2n)	key size (mn)	word bit size (n)	key word count (m)	const. seq.	rounds T
32	64	16	4	z_0	32
48	72	24	3	z_0	36
	96		4	z_1	36
64	96	32	3	z_2	42
	128		4	z_3	44
96	96	48	2	z_2	52
	144		3	z_3	54
128	128	64	2	z_2	68
	192		3	z_3	69
	256		4	z_4	72

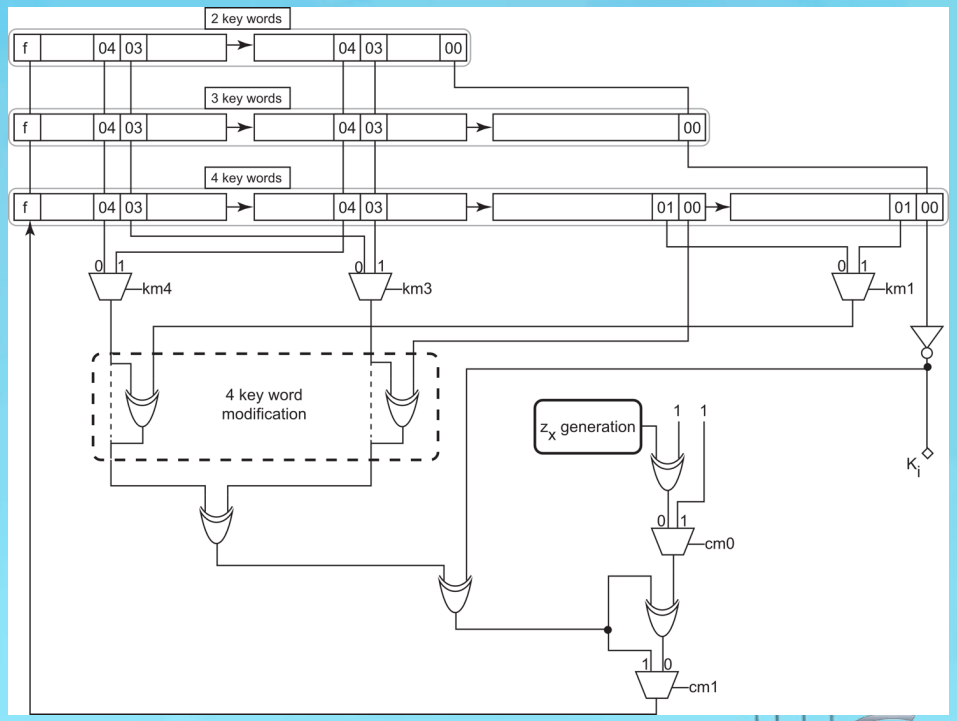
z_0 = 11111010001001010110000111001101111101000100101011000011100110

z_1 = 10001110111110010011000010110101000111011111001001100001011010

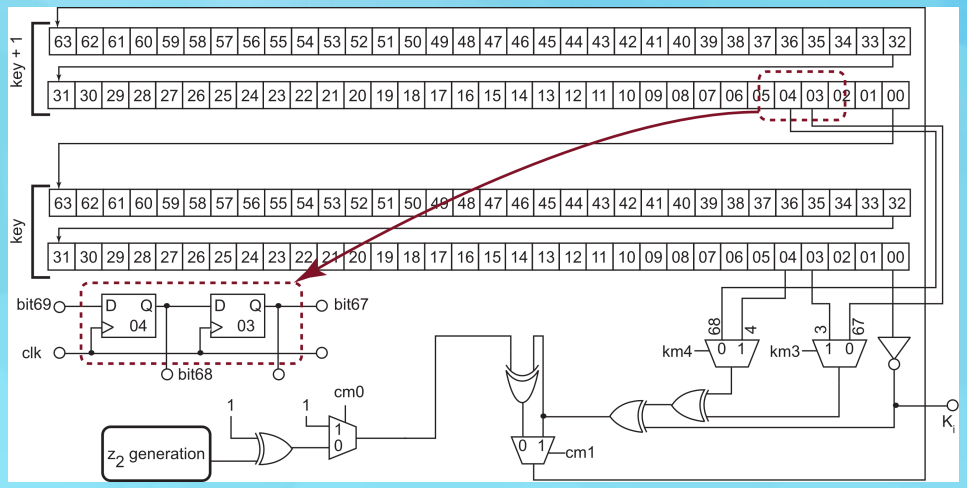
z_2 = 10101111011100000011010010011000101000010001111110010110110011

z_3 = 11011011101011000110010111100000010010001010011100110100001111

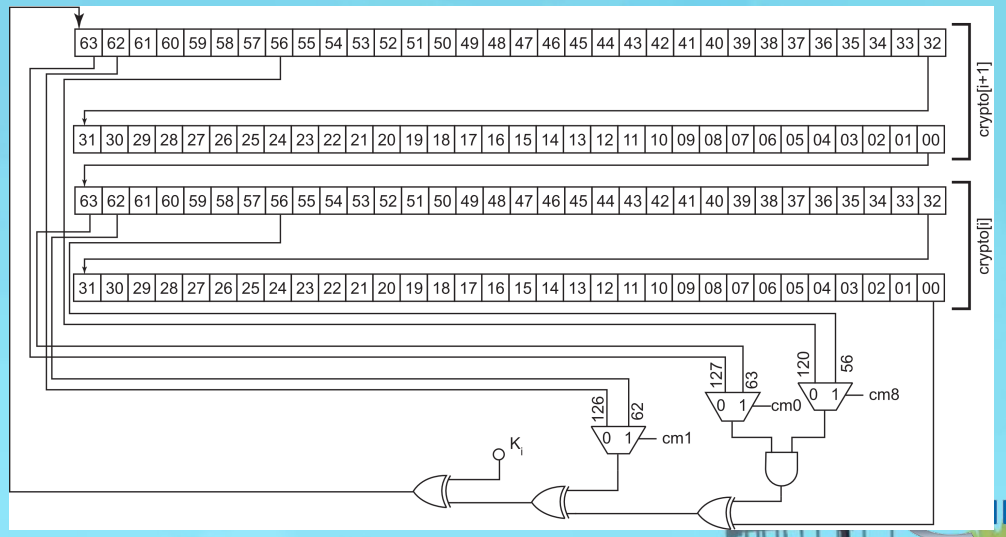
z_4 = 11010001111001101011011000100000010111000011001010010011101111



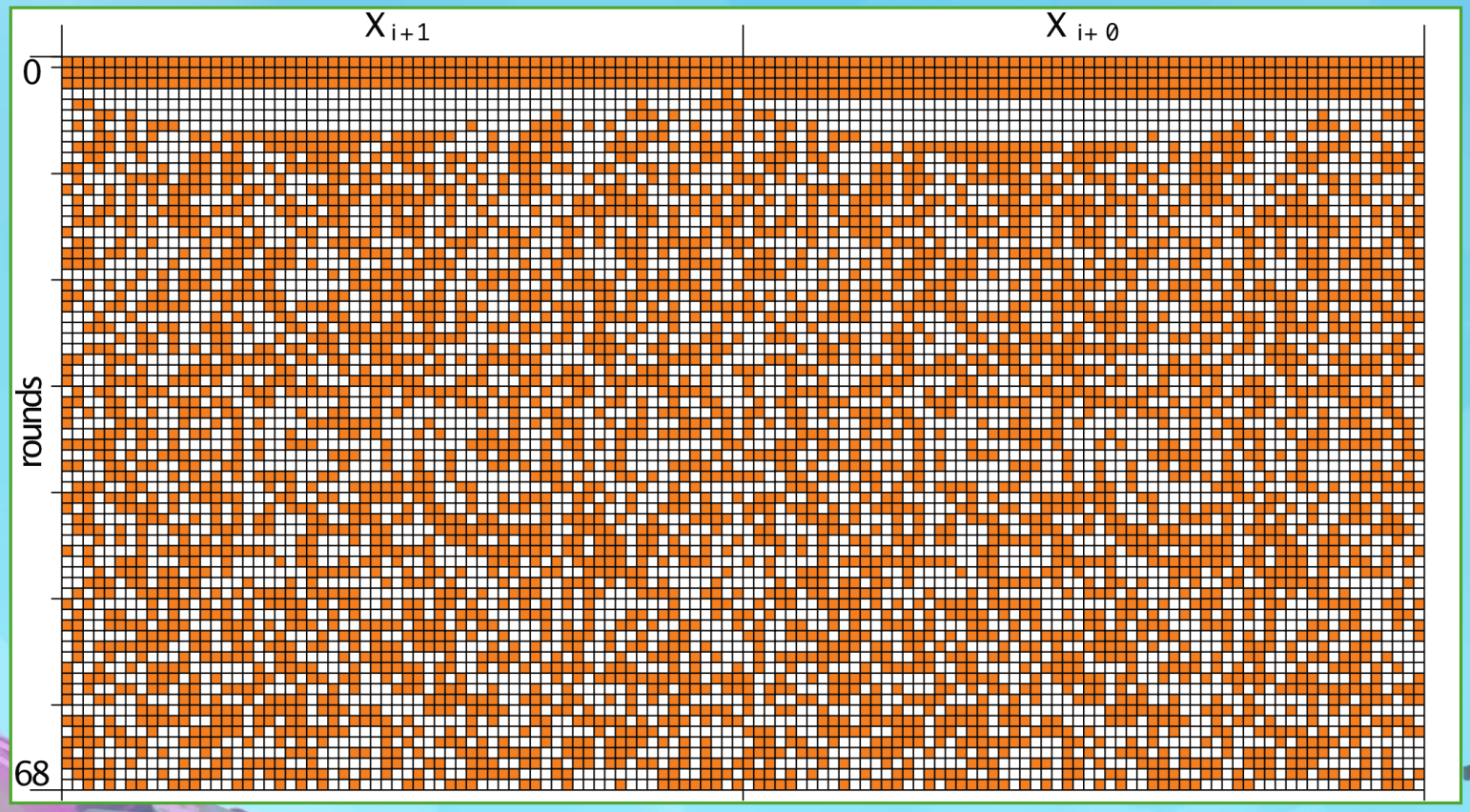
Ex. SIMON128/128: A “simple” symmetric cipher



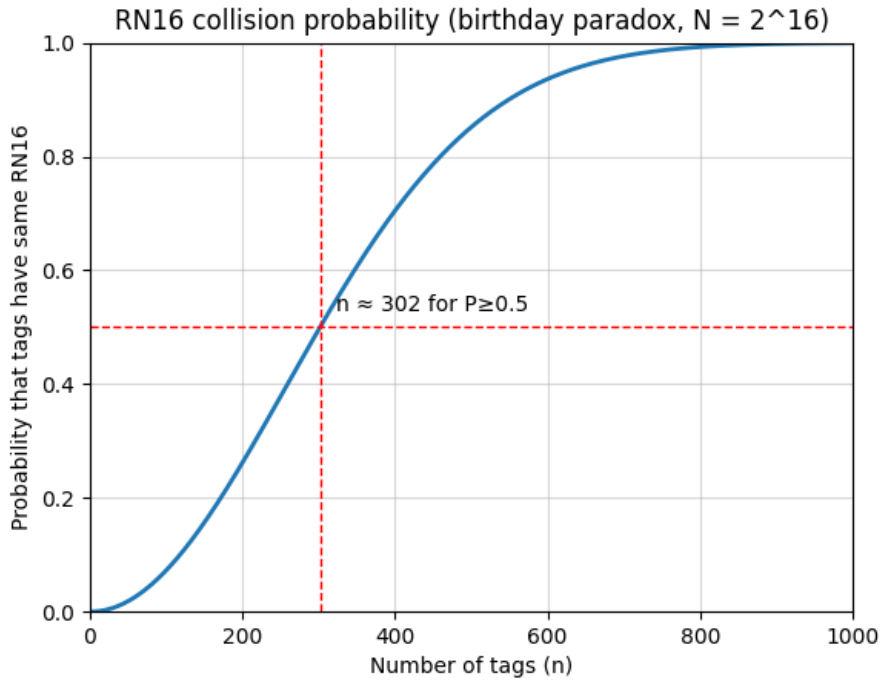
- 256 DFFs for encryption
- 12 DFFs for support

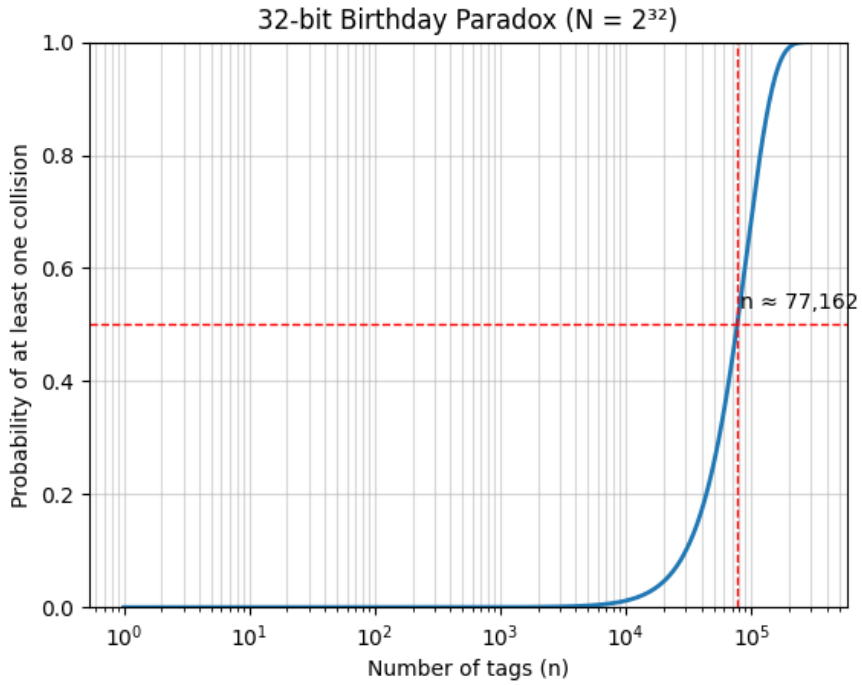


Ex. SIMON128/128: Key of 0 with data of 0



A few words on random numbers.





Why don’t we have strong cryptography on RFID?

- Cost (area)
- Cost (financial)
- Cost (time)
- Cost (power)
- Cost (software)
- Need?

Cryptography: Questions?

Monty Python's Life of Brian (1979)