



OPTIMIZING CHANNELIZATION FOR FUN AND PROFIT

Or: How I Learned to Stop Worrying and Love the Multiplexer

PREPARED FOR RAMS TRAINING SUMMIT XIII

NOVEMBER 30TH - DEC 1ST 2021

HUNTSVILLE, AL, US

BY GWYER SINCLAIR (NASA)

GWYER.Q.SINCLAIR@NASA.GOV

LEGAL / DISTRIBUTION NOTE

- Presentation is approved for public dissemination in present form only
- Presentation given by permission of NASA MSFC Export Control Office
- Presentation contains STI (Scientific Technical Information), subject to NASA Reporting, and identified by NF-1676 Document ID 20210024085
- The technology discussed is subject to an ongoing New Technology Report (NTR) titled *A Novel Algorithm for Multiplexer Channelization to Maximize Detection of Addressing Errors*
- NTR ID MFS-34250-1

WHAT'S THE BOTTOM LINE?

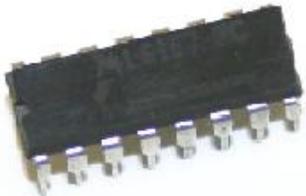
Design for reliability by using your multiplexer's un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to addressing failures.

WHAT'S THE BOTTOM LINE?

Design for reliability by using your multiplexer's un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to addressing failures.



"Free reliability?
In *my* unused multiplexer channels?"

It's more likely than you think.

[FREE PC CHECK!](#)

 CONTENTwatch

WHAT'S THE BOTTOM LINE?

Design for **reliability** by using your multiplexer's un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to addressing failures.

Reliability: The probability that a system will perform its intended function adequately for a specified period of time.

WHAT'S THE BOTTOM LINE?

Design for reliability by using your **multiplexer's** un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to addressing failures.

Multiplexer: A EEE device which selects between several input signals and forwards one (1) selected signal to output.
(Also known as a data selector or MUX)

WHAT'S THE BOTTOM LINE?

Design for reliability by using your multiplexer's **un-assigned channels**.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to addressing failures.

Un-assigned Channel: A MUX's input signal which has no sensor or device attached.

Un-assigned channels output GND or 0 V when selected. (channels 5-7 in example)

8 to 1 MUX	
Channel	Sensor
0	Pressure Sensor 1
1	Pressure Sensor 2
2	5V Voltage Reference
3	Temp. Sensor 1
4	Temp. Sensor 2
5	N/A
6	N/A
7	N/A

WHAT'S THE BOTTOM LINE?

Design for reliability by using your multiplexer's un-assigned channels.

With optimal **channelization**, software can detect, correct, and eliminate critical outcomes due to addressing failures.

Channelization: The order in which sensors/signals are connected to the MUX inputs

8 to 1 MUX	
Channel	Sensor
0	Pressure Sensor 1
1	Pressure Sensor 2
2	5V Voltage Reference
3	Temp. Sensor 1
4	Temp. Sensor 2
5	N/A
6	N/A
7	N/A

WHAT'S THE BOTTOM LINE?

Design for reliability by using your multiplexer's un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to **addressing failures.**

Addressing Failure: a failure in the software, logic circuit, or MUX itself which causes the wrong channel to be requested or transmitted.

8 to 1 MUX	
Channel	Sensor
0	Pressure Sensor 1
1	Pressure Sensor 2
2	5V Voltage Reference
3	Temp. Sensor 1
4	Temp. Sensor 2
5	N/A
6	N/A
7	N/A

WHAT'S THE BOTTOM LINE?

Design for reliability by using your multiplexer's un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to **addressing failures.**

Addressing Failure: a failure in the software, logic circuit, or MUX itself which causes the wrong channel to be requested or transmitted.

WANT

8 to 1 MUX	
Channel	Sensor
0	Pressure Sensor 1
1	Pressure Sensor 2
2	5V Voltage Reference
3	Temp. Sensor 1
4	Temp. Sensor 2
5	N/A
6	N/A
7	N/A

WHAT'S THE BOTTOM LINE?

Design for reliability by using your multiplexer's un-assigned channels.

With optimal channelization, software can detect, correct, and eliminate critical outcomes due to **addressing failures.**

Addressing Failure: a failure in the software, logic circuit, or MUX itself which causes the wrong channel to be requested or transmitted.

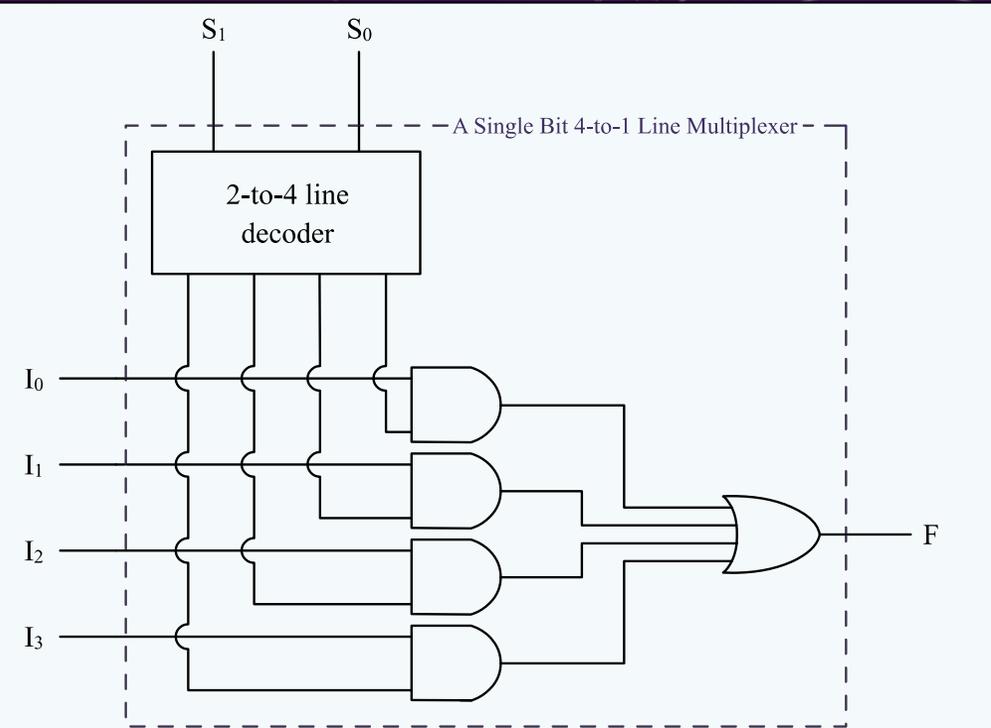
GET

8 to 1 MUX	
Channel	Sensor
0	Pressure Sensor 1
1	Pressure Sensor 2
2	5V Voltage Reference
3	Temp. Sensor 1
4	Temp. Sensor 2
5	N/A
6	N/A
7	N/A

MULTIPLEXER MECHANICS

A MUX uses 'select lines' and logic gates to control which input is sent to output. In a 4-to-1 MUX there are 2 select lines (S0, S1). Select digits form the channel # in binary.

4 to 1 MUX		
Channel	S1	S0
0	0	0
1	0	1
2	1	0
3	1	1

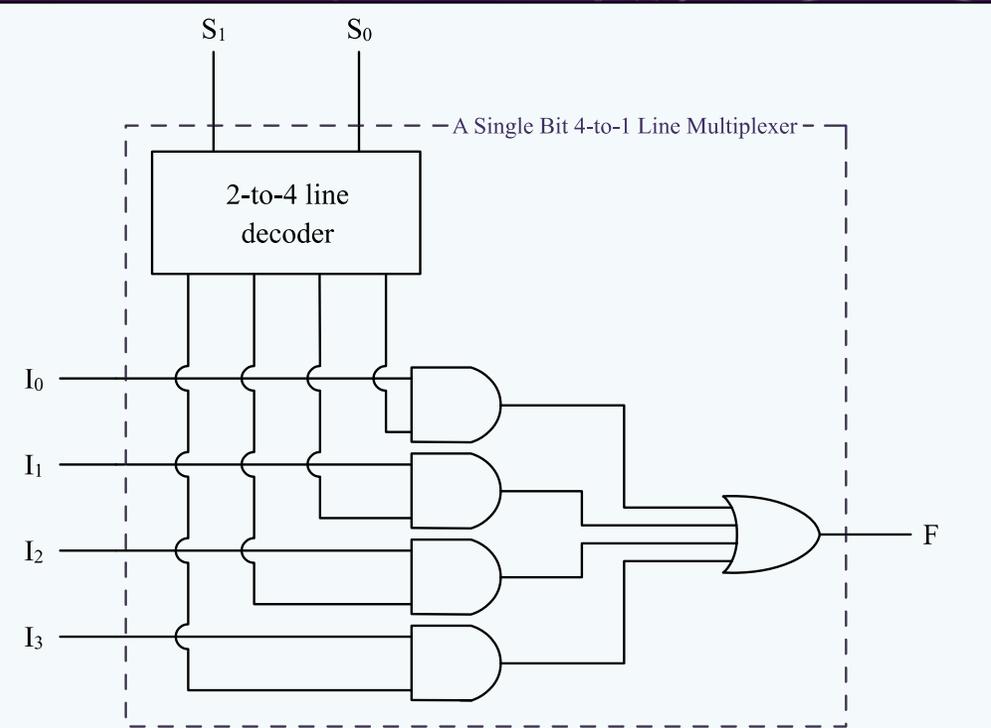


Truth Table

S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F
0 0	0	0	0	0	0	0 1	0	0	0	0	0	1 0	0	0	0	0	0	1 1	0	0	0	0	0
	0	0	0	1	1		0	0	0	1	0		0	0	0	1	0		0	0	0	1	0
	0	0	1	0	0		0	0	1	0	1		0	0	1	0	0		0	0	1	0	0
	0	0	1	1	1		0	0	1	1	1		0	0	1	1	0		0	0	1	1	0
	0	1	0	0	0		0	1	0	0	0		0	1	0	0	1		0	1	0	0	0
	0	1	0	1	1		0	1	0	1	0		0	1	0	1	1		0	1	0	1	0
	0	1	1	0	0		0	1	1	0	1		0	1	1	0	1		0	1	1	0	0
	0	1	1	1	1		0	1	1	1	1		0	1	1	1	1		0	1	1	1	0
	1	0	0	0	0		1	0	0	0	0		1	0	0	0	0		1	0	0	0	1
	1	0	0	1	1		1	0	0	1	0		1	0	0	1	0		1	0	0	1	1
	1	0	1	0	0		1	0	1	0	1		1	0	1	0	0		1	0	1	0	1
	1	0	1	1	1		1	0	1	1	1		1	0	1	1	0		1	0	1	1	1
	1	1	0	0	0		1	1	0	0	0		1	1	0	0	1		1	1	0	0	1
	1	1	0	1	1		1	1	0	1	0		1	1	0	1	1		1	1	0	1	1
	1	1	1	0	0		1	1	1	0	1		1	1	1	0	1		1	1	1	0	1
	1	1	1	1	1		1	1	1	1	1		1	1	1	1	1		1	1	1	1	1

ADDRESSING FAILURE MECHANICS (MUX CAUSE)

A short (unintended connection) between any number of input or select lines will force them to share the same voltage.



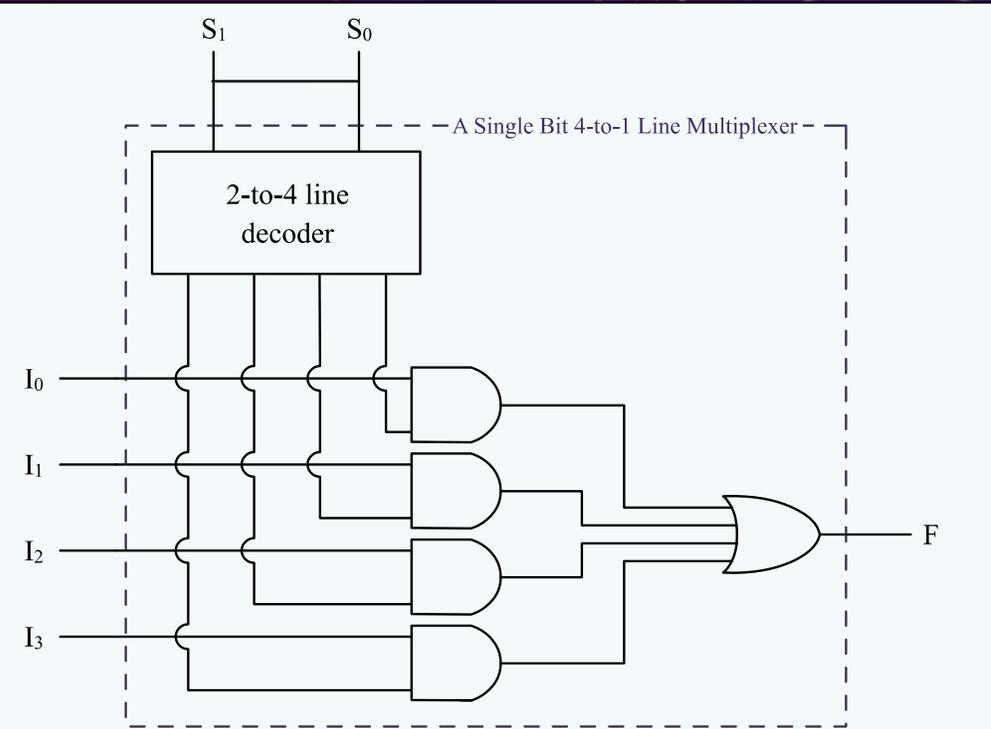
Truth Table

S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁ S ₀	I ₃	I ₂	I ₁	I ₀	F
0 0	0	0	0	0	0	0 0	0	0	0	0	0	0 0	0	0	0	0	0	0 0	0	0	0	0	0
	0	0	0	1	1		0	0	0	1	0		0	0	0	1	0		0	0	0	1	0
	0	0	1	0	0		0	0	1	0	1		0	0	1	0	0		0	0	1	0	0
	0	0	1	1	1		0	0	1	1	1		0	0	1	1	0		0	0	1	1	0
	0	1	0	0	0		0	1	0	0	0		0	1	0	0	1		0	1	0	0	0
	0	1	0	1	1		0	1	0	1	0		0	1	0	1	1		0	1	0	1	0
	0	1	1	0	0		0	1	1	0	1		0	1	1	0	1		0	1	1	0	0
	0	1	1	1	1		0	1	1	1	1		0	1	1	1	1		0	1	1	1	0
	1	0	0	0	0		1	0	0	0	0		1	0	0	0	0		1	0	0	0	1
	1	0	0	1	1		1	0	0	1	0		1	0	0	1	0		1	0	0	1	1
	1	0	1	0	0		1	0	1	0	1		1	0	1	0	0		1	0	1	0	1
	1	0	1	1	1		1	0	1	1	1		1	0	1	1	0		1	0	1	1	1
	1	1	0	0	0		1	1	0	0	0		1	1	0	0	1		1	1	0	0	1
	1	1	0	1	1		1	1	0	1	0		1	1	0	1	1		1	1	0	1	1
	1	1	1	0	0		1	1	1	0	1		1	1	1	0	1		1	1	1	0	1
	1	1	1	1	1		1	1	1	1	1		1	1	1	1	1		1	1	1	1	1

ADDRESSING FAILURE MECHANICS (MUX CAUSE)

Here, a short between S1 and S0 will cause both lines to be '1' if either is '1'

Channels 1 and 2 are unreachable states



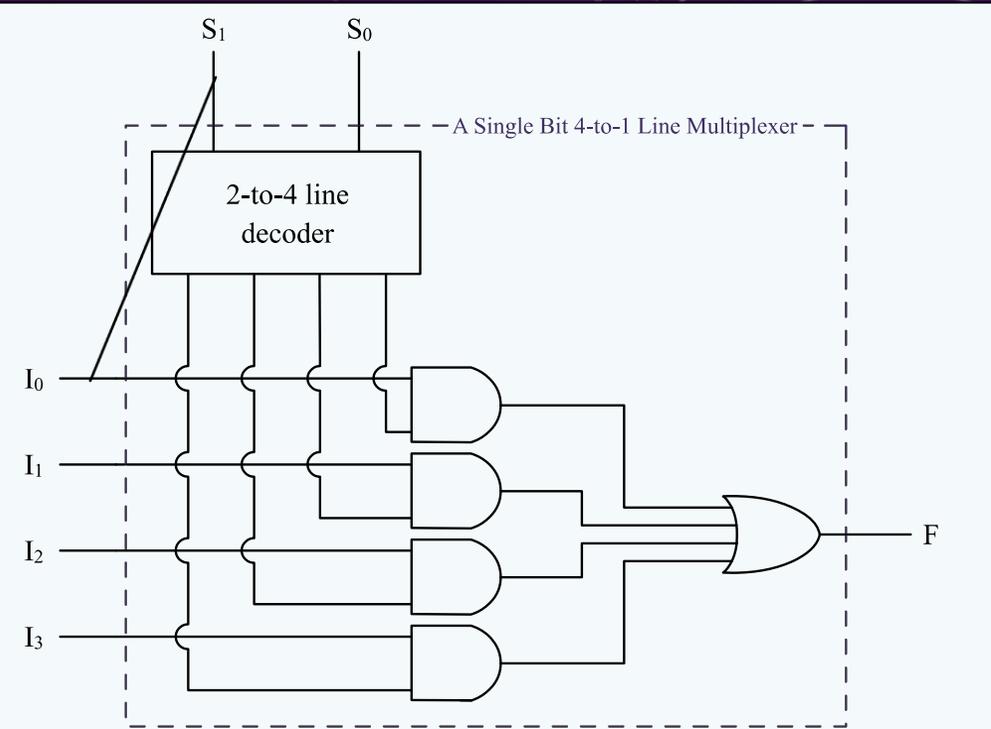
Truth Table

S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
		0	0	0	1	1			0	0	0	1	0			0	0	0	1	0	0
		0	0	1	0	0			0	0	1	0	1			0	0	1	0	0	0
		0	0	1	1	1			0	0	1	1	1			0	0	1	1	0	0
		0	1	0	0	0			0	1	0	0	0			0	1	0	0	1	0
		0	1	0	1	1			0	1	0	1	0			0	1	0	1	1	0
		0	1	1	0	0			0	1	1	0	1			0	1	1	0	1	0
		0	1	1	1	1			0	1	1	1	1			0	1	1	1	1	0
		1	0	0	0	0			1	0	0	0	0			1	0	0	0	0	1
		1	0	0	1	1			1	0	0	1	0			1	0	0	1	0	1
		1	0	1	0	0			1	0	1	0	1			1	0	1	0	0	1
		1	0	1	1	1			1	0	1	1	1			1	0	1	1	0	1
		1	1	0	0	0			1	1	0	0	0			1	1	0	0	1	1
		1	1	0	1	1			1	1	0	1	0			1	1	0	1	1	1
		1	1	1	0	0			1	1	1	0	1			1	1	1	0	1	1
		1	1	1	1	1			1	1	1	1	1			1	1	1	1	1	1

ADDRESSING FAILURE MECHANICS (MUX CAUSE)

Here, a short between S1 and I0 (A grounded 0V reference line) will cause S1 to always be '0' or to 'flicker'

Channels 2 and 3 are unreliable states

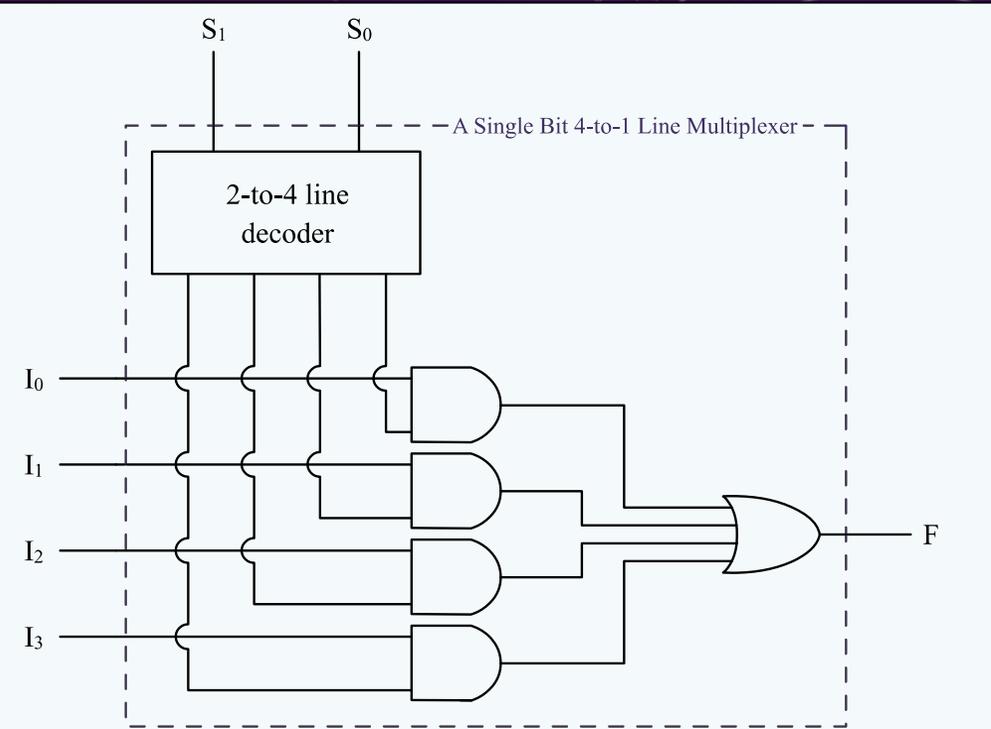


Truth Table

S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	0	1	1	1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0
0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1
1	0	0	1	1	1	1	0	1	0	0	1	0	0	0	1	0	0	1	1	0	1
1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1
1	0	1	1	1	1	1	0	1	0	1	1	1	0	0	1	0	1	1	1	0	1
1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1
1	1	0	1	1	1	1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	1
1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	1
1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	1	1	1	1	0	1

ADDRESSING FAILURE MECHANICS (MUX CAUSE)

Any failure, erratic, or unintended behavior from EEE elements like the logic gates can cause unpredictable channel selection



Truth Table

S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	S ₁	S ₀	I ₃	I ₂	I ₁	I ₀	F	
0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
		0	0	0	1	1			0	0	0	1	0			0	0	0	1	0			0	0	0	1	0	
		0	0	1	0	0			0	0	1	0	1			0	0	1	0	0			0	0	1	0	0	
		0	0	1	1	1			0	0	1	1	1			0	0	1	1	0			0	0	1	1	0	
		0	1	0	0	0			0	1	0	0	0			0	1	0	0	1			0	1	0	0	0	
		0	1	0	1	1			0	1	0	1	0			0	1	0	1	1			0	1	0	1	0	
		0	1	1	0	0			0	1	1	0	1			0	1	1	0	1			0	1	1	0	0	
		0	1	1	1	1			0	1	1	1	1			0	1	1	1	1			0	1	1	1	0	
		1	0	0	0	0			1	0	0	0	0			1	0	0	0	0			1	0	0	0	1	
		1	0	0	1	1			1	0	0	1	0			1	0	0	1	0			1	0	0	1	1	
		1	0	1	0	0			1	0	1	0	1			1	0	1	0	0			1	0	1	0	1	
		1	0	1	1	1			1	0	1	1	1			1	0	1	1	0			1	0	1	1	1	
		1	1	0	0	0			1	1	0	0	0			1	1	0	0	1			1	1	0	0	1	
		1	1	0	1	1			1	1	0	1	0			1	1	0	1	1			1	1	0	1	1	
		1	1	1	0	0			1	1	1	0	1			1	1	1	0	1			1	1	1	0	1	
		1	1	1	1	1			1	1	1	1	1			1	1	1	1	1			1	1	1	1	1	

ADDRESSING FAILURE MECHANICS (HARDWARE CAUSE)

- A short involving non-MUX components
- A short or sneak path which circumvents the MUX
- Blown res. or cap. causes open circuit ahead of a select line
- Short to voltage or signal source ahead of a select line
- IC manufacturing defects

ADDRESSING FAILURE MECHANICS (SOFTWARE CAUSE)

- Memory or buffer errors
- Hackers
- Firewalls
- Trojans
- URLs
- Phishing Emails
- The Three Laws of Robotics

ADDRESSING FAILURE MECHANICS (SOFTWARE CAUSE)

(Seriously)

Any error which causes software to request the wrong channel introduces an addressing error into a circuit without hardware errors.

ADDRESSING FAILURE MECHANICS - EXAMPLE

Let's return to the example channelization shown in the introduction.

This 8-to-1 MUX has:

- 2 pressure sensors
- 2 temperature sensors,
- 5V voltage reference,
- 3 unused channels

8 to 1 MUX				
Channel	A2	A1	A0	Sensor
0	0	0	0	Pressure Sensor 1
1	0	0	1	Pressure Sensor 2
2	0	1	0	5V Voltage Reference
3	0	1	1	Temp. Sensor 1
4	1	0	0	Temp. Sensor 2
5	1	0	1	N/A
6	1	1	0	N/A
7	1	1	1	N/A

ADDRESSING FAILURE MECHANICS - EXAMPLE

Consider an electrical short which causes A2 to be stuck 'high' (always have logic '1', regardless of command)

Expected					Received				
Channel	A2	A1	A0		Channel	A2	A1	A0	
0	0	0	0	Pressure Sensor 1	4	1	0	0	Temp. Sensor2
1	0	0	1	Pressure Sensor 2	5	1	0	1	N/A
2	0	1	0	5V Voltage Reference	6	1	1	0	N/A
3	0	1	1	Temp. Sensor 1	7	1	1	1	N/A
4	1	0	0	Temp. Sensor2	4	1	0	0	Temp. Sensor2
5	1	0	1	N/A	5	1	0	1	N/A
6	1	1	0	N/A	6	1	1	0	N/A
7	1	1	1	N/A	7	1	1	1	N/A

ADDRESSING FAILURE MECHANICS - EXAMPLE

Now, when expecting telemetry from channel 0, values from channel 4 are transmitted instead. When channels 1-3 are expected, the transmitted signal is from channels 5-7 instead. N/A channels output 0V.

Expected					Received				
Channel	A2	A1	A0		Channel	A2	A1	A0	
0	0	0	0	Pressure Sensor 1	4	1	0	0	Temp. Sensor2
1	0	0	1	Pressure Sensor 2	5	1	0	1	N/A
2	0	1	0	5V Voltage Reference	6	1	1	0	N/A
3	0	1	1	Temp. Sensor 1	7	1	1	1	N/A
4	1	0	0	Temp. Sensor2	4	1	0	0	Temp. Sensor2
5	1	0	1	N/A	5	1	0	1	N/A
6	1	1	0	N/A	6	1	1	0	N/A
7	1	1	1	N/A	7	1	1	1	N/A

ADDRESSING FAILURE MECHANICS - EXAMPLE

Consider a launch vehicle using a pressure sensor suite to determine pressure in its propellant tanks. With the wrong information telemetered, the vehicle underestimates the pressure, and software decides to increase it.

Now we have a critical overpressure condition!

Expected					Received				
Channel	A2	A1	A0		Channel	A2	A1	A0	
0	0	0	0	Pressure Sensor 1	4	1	0	0	Temp. Sensor2
1	0	0	1	Pressure Sensor 2	5	1	0	1	N/A
2	0	1	0	5V Voltage Reference	6	1	1	0	N/A
3	0	1	1	Temp. Sensor 1	7	1	1	1	N/A
4	1	0	0	Temp. Sensor2	4	1	0	0	Temp. Sensor2
5	1	0	1	N/A	5	1	0	1	N/A
6	1	1	0	N/A	6	1	1	0	N/A
7	1	1	1	N/A	7	1	1	1	N/A

DETECT THESE ERRORS WITH CHANNELIZATION

Channels 1-3 are now erroneously telemetering data from channels 5-7.

Since these are un-assigned channels, this is a full scale low – 0 Volts.

A full scale low reading can be detected with software and disqualified.

Expected					Received				
Channel	A2	A1	A0		Channel	A2	A1	A0	
0	0	0	0	Pressure Sensor 1	4	1	0	0	Temp. Sensor2
1	0	0	1	Pressure Sensor 2	5	1	0	1	N/A
2	0	1	0	5V Voltage Reference	6	1	1	0	N/A
3	0	1	1	Temp. Sensor 1	7	1	1	1	N/A
4	1	0	0	Temp. Sensor2	4	1	0	0	Temp. Sensor2
5	1	0	1	N/A	5	1	0	1	N/A
6	1	1	0	N/A	6	1	1	0	N/A
7	1	1	1	N/A	7	1	1	1	N/A

DETECT THESE ERRORS WITH CHANNELIZATION

Depending on your product or program's sensor suite and software, this may mean that the erroneous measurement can be detected and excluded from use, or corrected, *before* the failure propagates.

Expected					Received				
Channel	A2	A1	A0		Channel	A2	A1	A0	
0	0	0	0	Pressure Sensor 1	4	1	0	0	Temp. Sensor2
1	0	0	1	Pressure Sensor 2	5	1	0	1	N/A
2	0	1	0	5V Voltage Reference	6	1	1	0	N/A
3	0	1	1	Temp. Sensor 1	7	1	1	1	N/A
4	1	0	0	Temp. Sensor2	4	1	0	0	Temp. Sensor2
5	1	0	1	N/A	5	1	0	1	N/A
6	1	1	0	N/A	6	1	1	0	N/A
7	1	1	1	N/A	7	1	1	1	N/A

DETECT THESE ERRORS WITH CHANNELIZATION

This is possible due to the different expected ranges of each channel.

- Sensors each return a voltage which is interpreted by software
- Different makes, models, and use cases of sensors have different exp. voltage ranges
- Reference or constant voltage signals will always have a known value
- Unassigned channels will have a voltage of GND or 0V

When a system suffers an addressing error, it may or may not receive a signal within the expected range. A signal outside the expected range can be detected with hardware (voltage based switching) or a software solution.

DETECT THESE ERRORS WITH CHANNELIZATION

Can we channelize these sensors in such a way that an addressing failure always causes the MUX to telemeter a detectably incorrect channel?



LET'S ASK RICHARD!

- Richard Hamming (1915 – 1998)
- American mathematician
- CPE and Telecom : Hamming code, Hamming window, sphere-packing, Hamming Distance.
- Manhattan Project

“The purpose of computing is insight, not numbers”

Photo : [Wikipedia via Bell Labs](#)

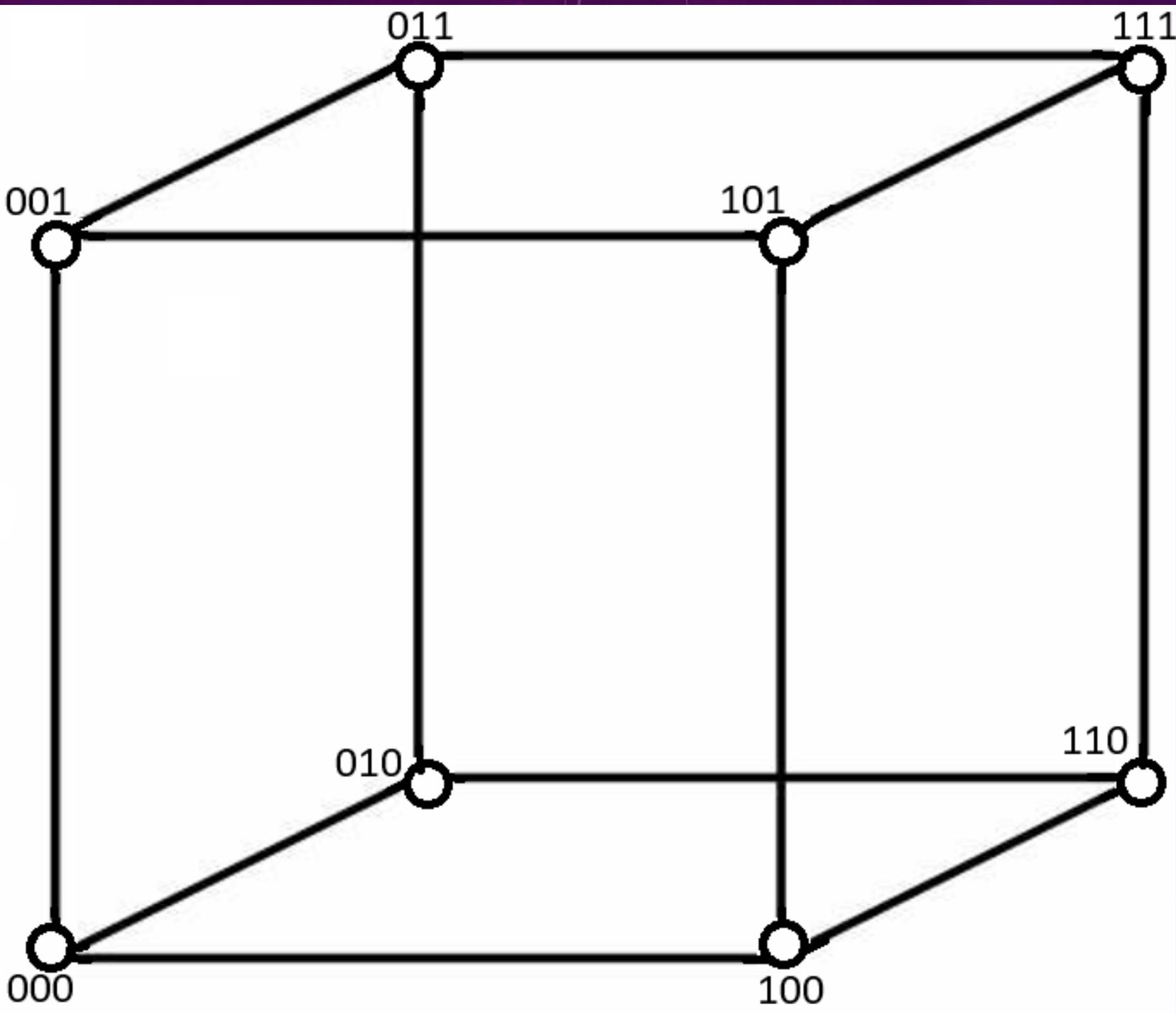
HAMMING DISTANCE

The Hamming Distance between two binary expressions of equal length is the number of bit positions in which the two bits are different.

Expression A			Expression B			Hamming Distance
0	0	0	0	0	0	0
0	0	0	0	1	1	2
0	1	0	1	0	1	3

In an 8-to-1 MUX (3 binary digits), a single addressing failure means that the expected channel could be replaced by **any of the channels within a Hamming Distance of 1**. I will call these channels Hamming-Adjacent.

100 -> 000, 110, or 101



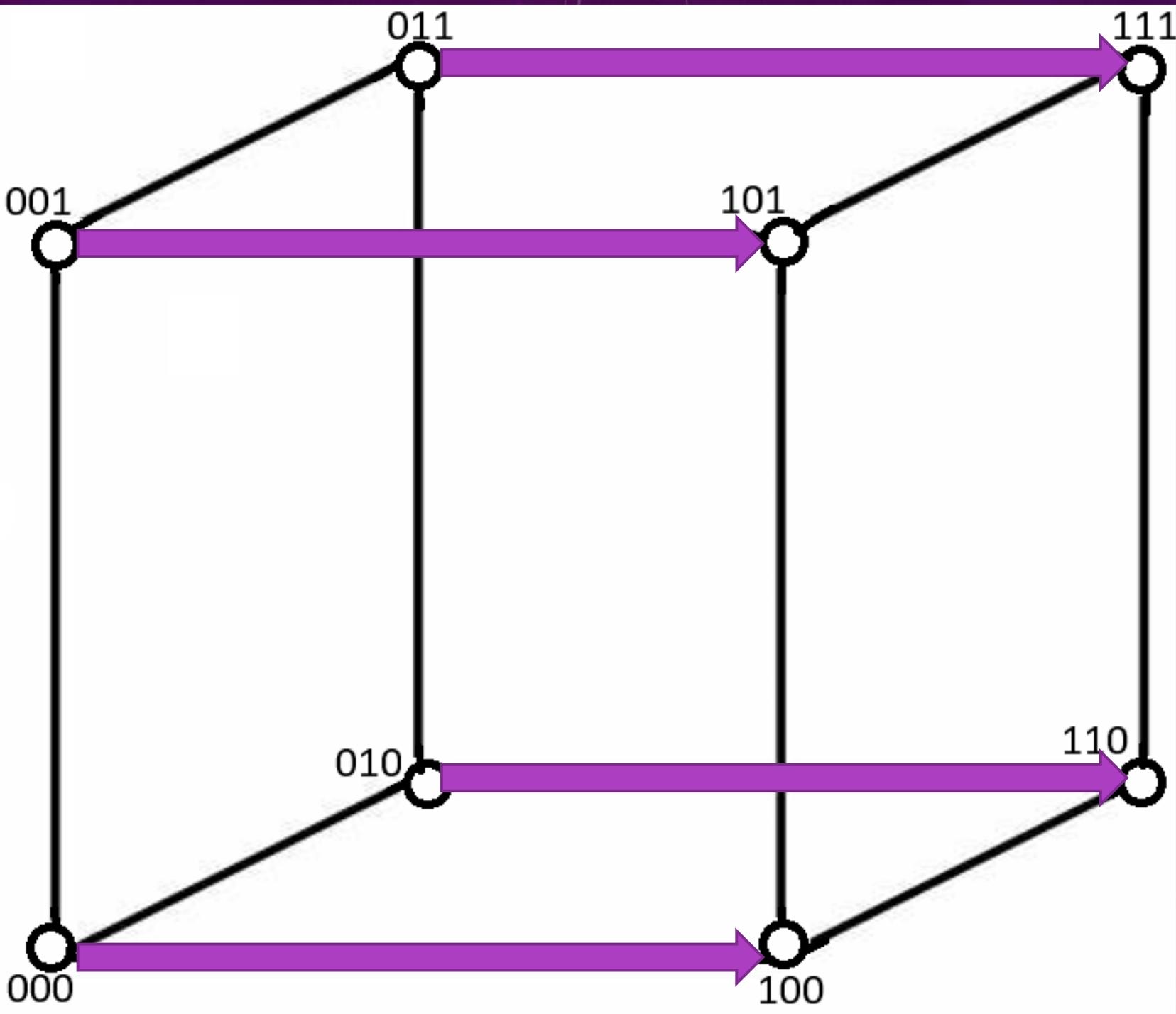
HAMMING CUBE

8 POSITIONS

The Hamming Cube is a visual representation of the addressing space of an 8 position MUX.

The Hamming Cube can be viewed as the unit cube in cartesian space. $(0,0,0)$ is at the origin, and each vertex is a distance of 1 in X, Y, or Z.

The axes correspond to select lines A0, A1, and A2



HAMMING CUBE

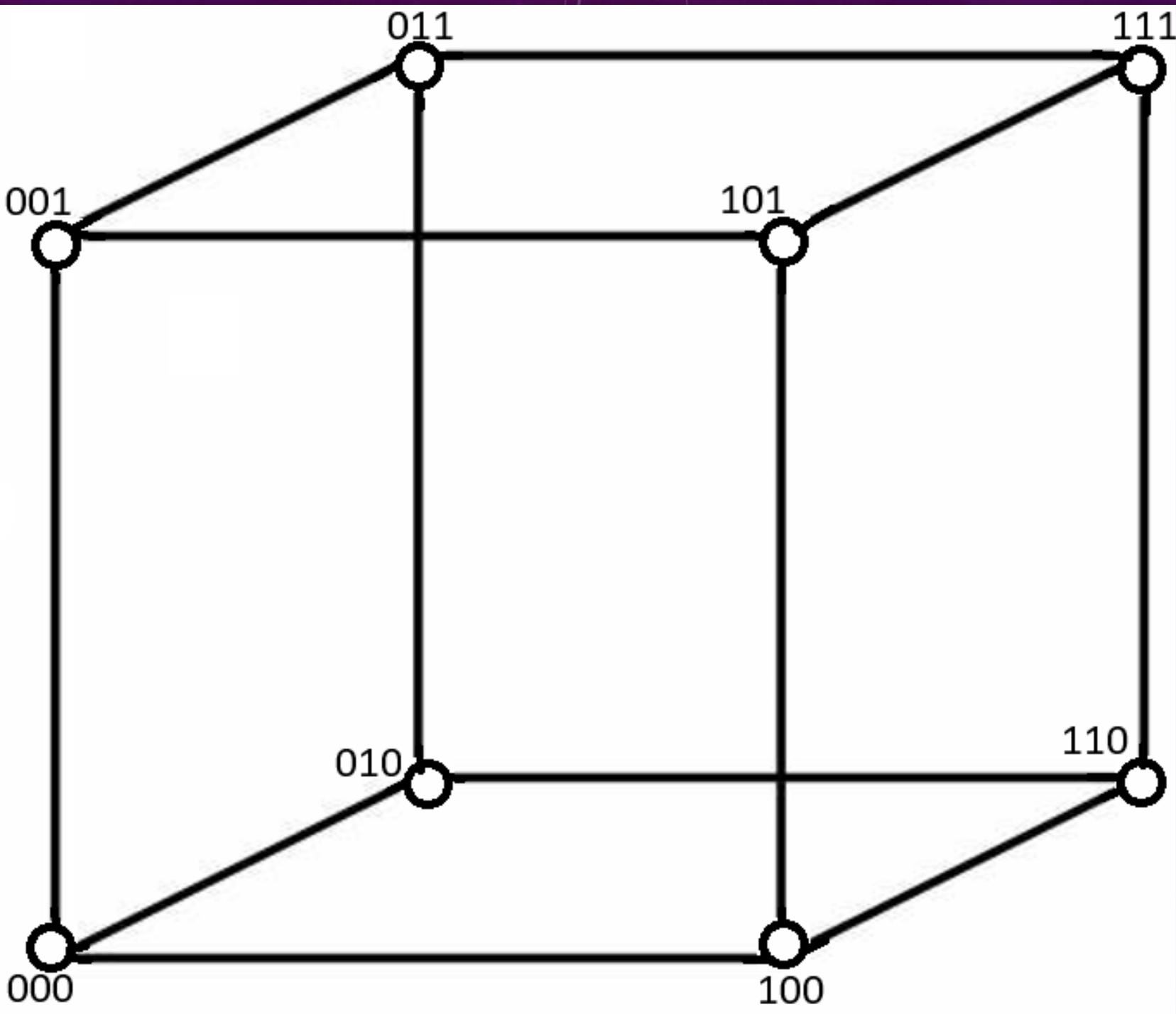
8 POSITIONS

Visualize the addressing failure from our example, A2 (leading bit) stuck high.

000 fails to 100

001 fails to 101

Etc...

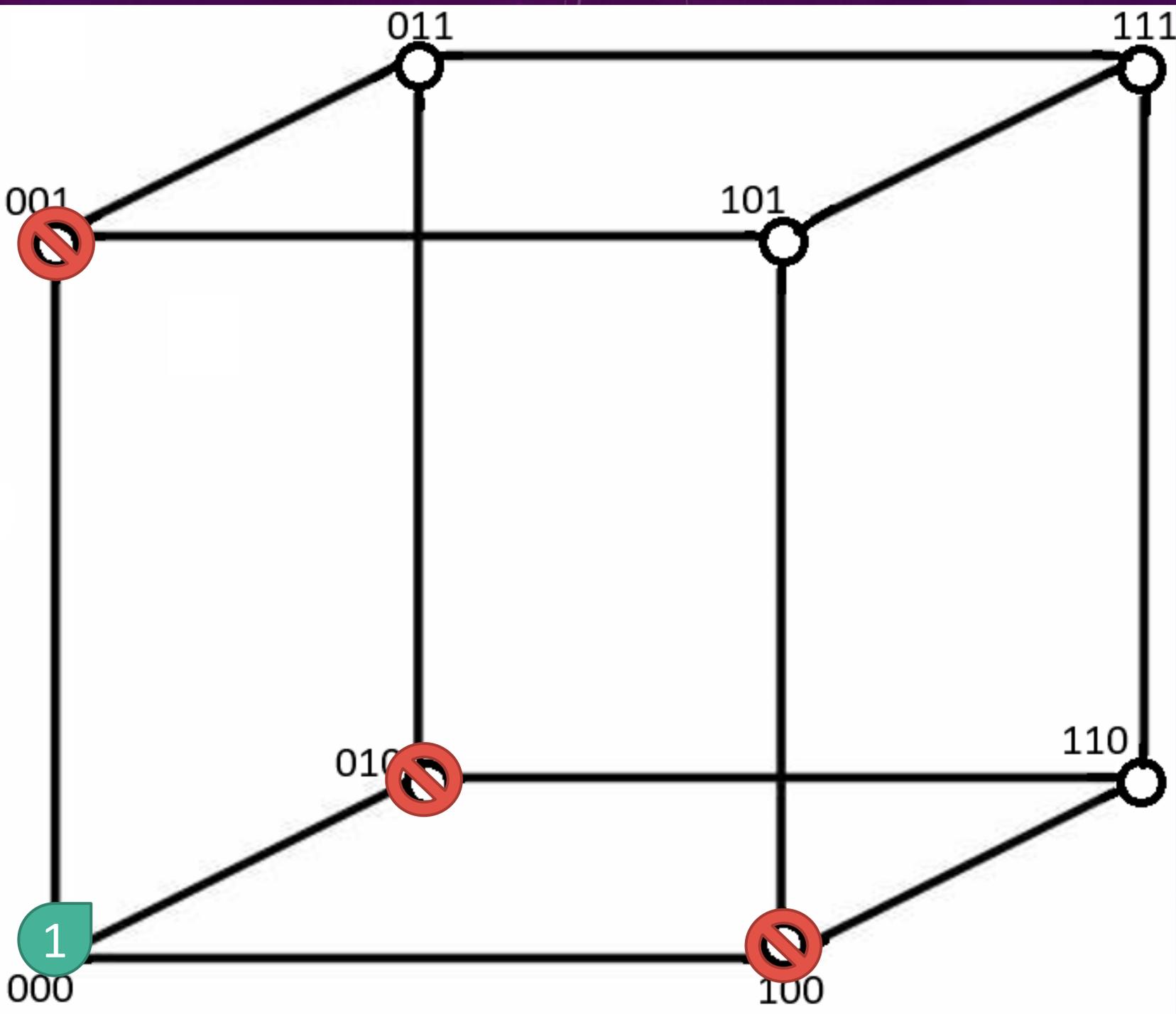


HAMMING CUBE

8 POSITIONS

Any two vertices on this cube which are connected by an edge have a Hamming Distance of 1. (Hamming Adjacent)

In the event of an addressing failure, they would be susceptible 'fail-over' channels.

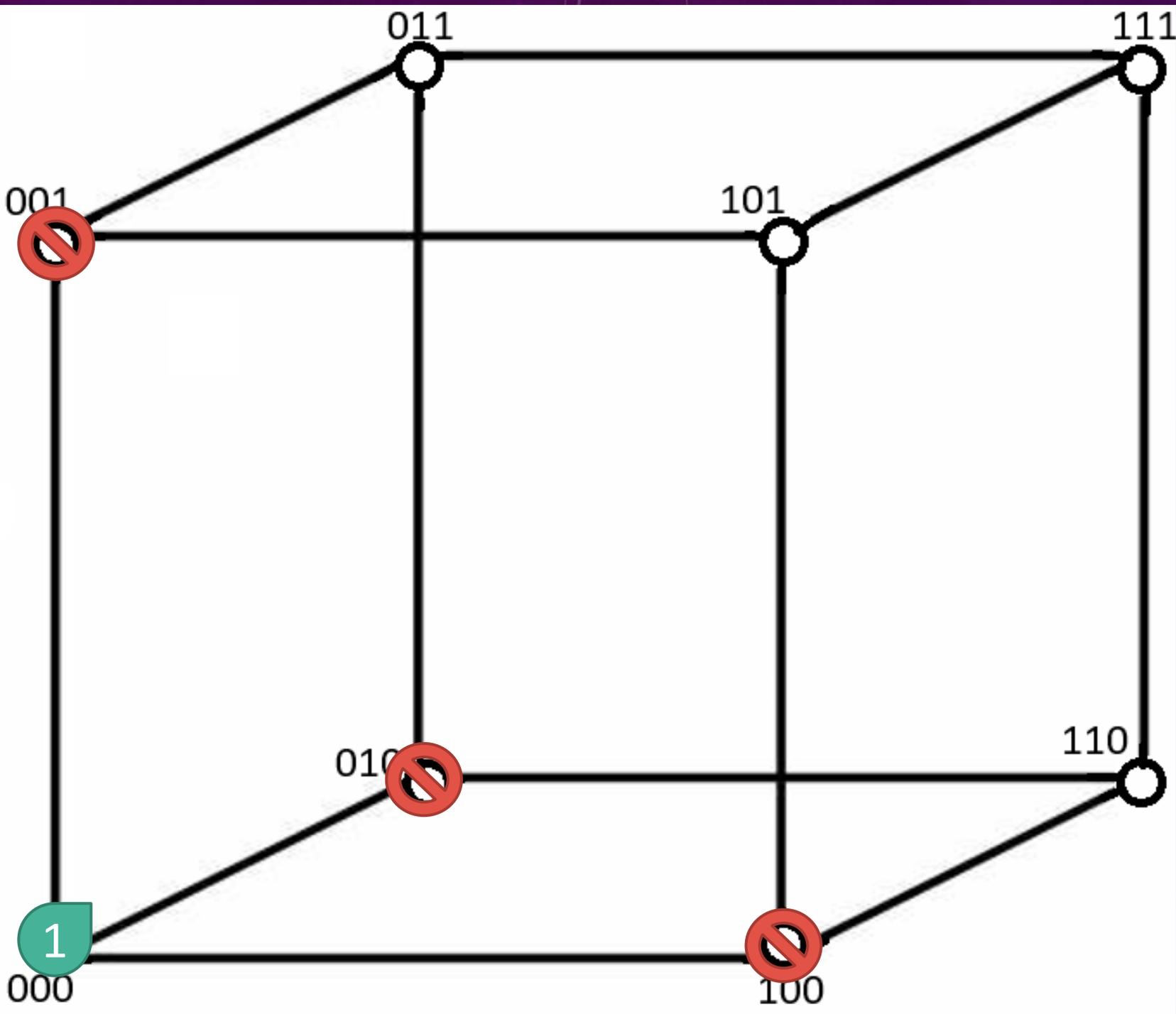


HAMMING CUBE

8 POSITIONS

Let's use this cube to place signals that cannot fail-over into another used channel.

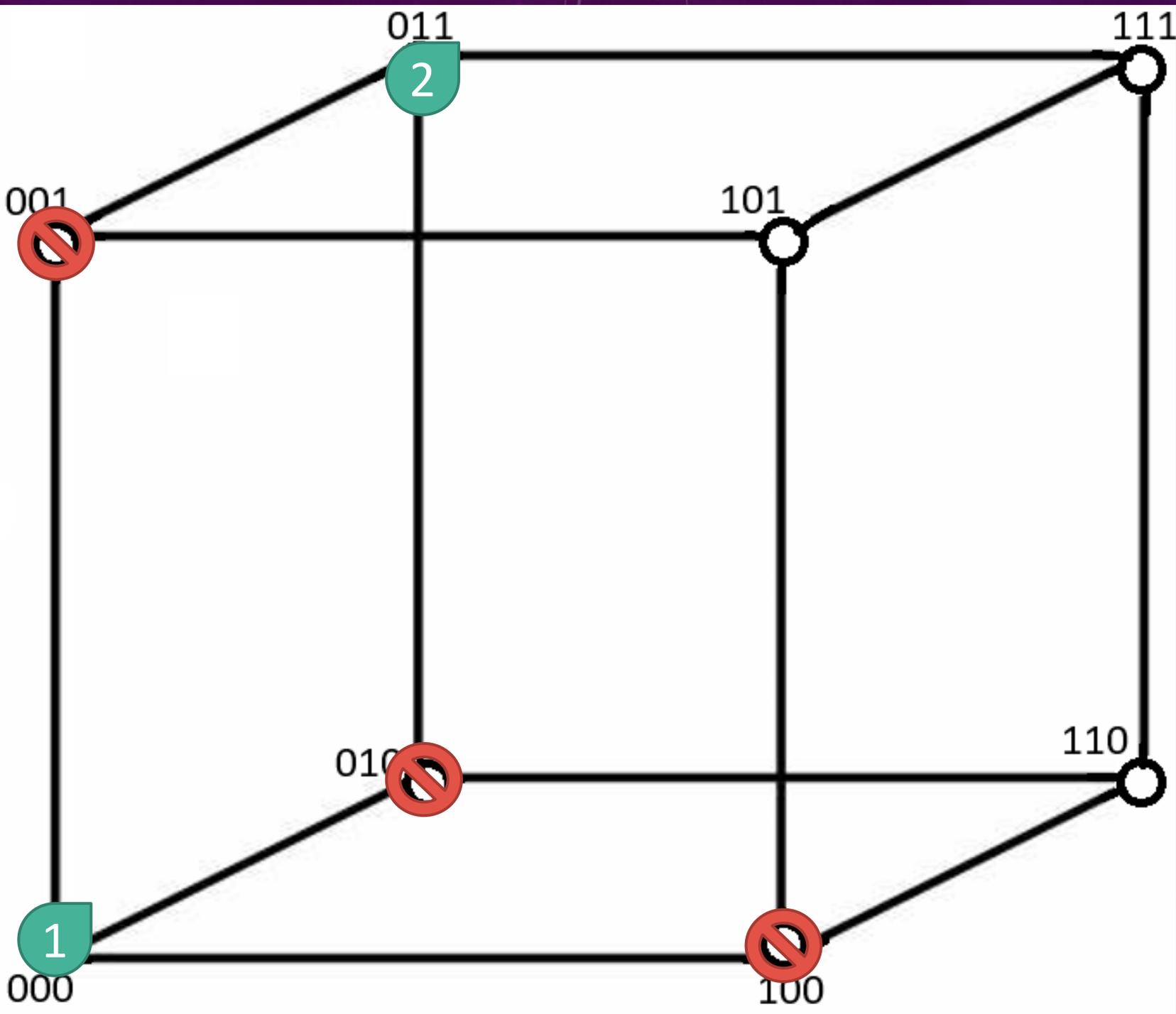
If we begin by placing one of our signals in 000, that means 001, 010, and 100 should be kept un-assigned.



HAMMING CUBE

8 POSITIONS

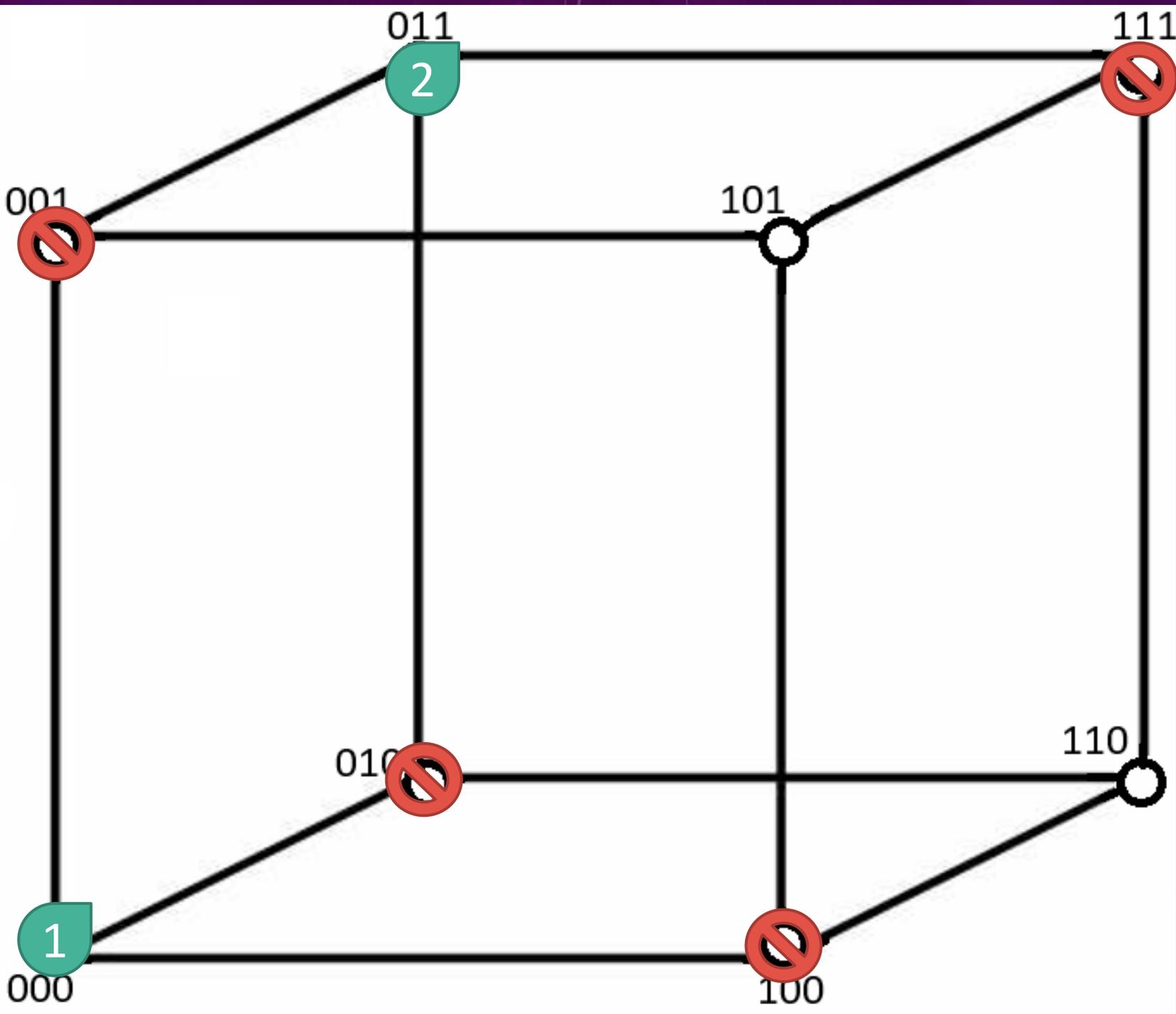
So on, fill the next available channel and reserve each channel within a hamming distance of 1 as un-assigned



HAMMING CUBE

8 POSITIONS

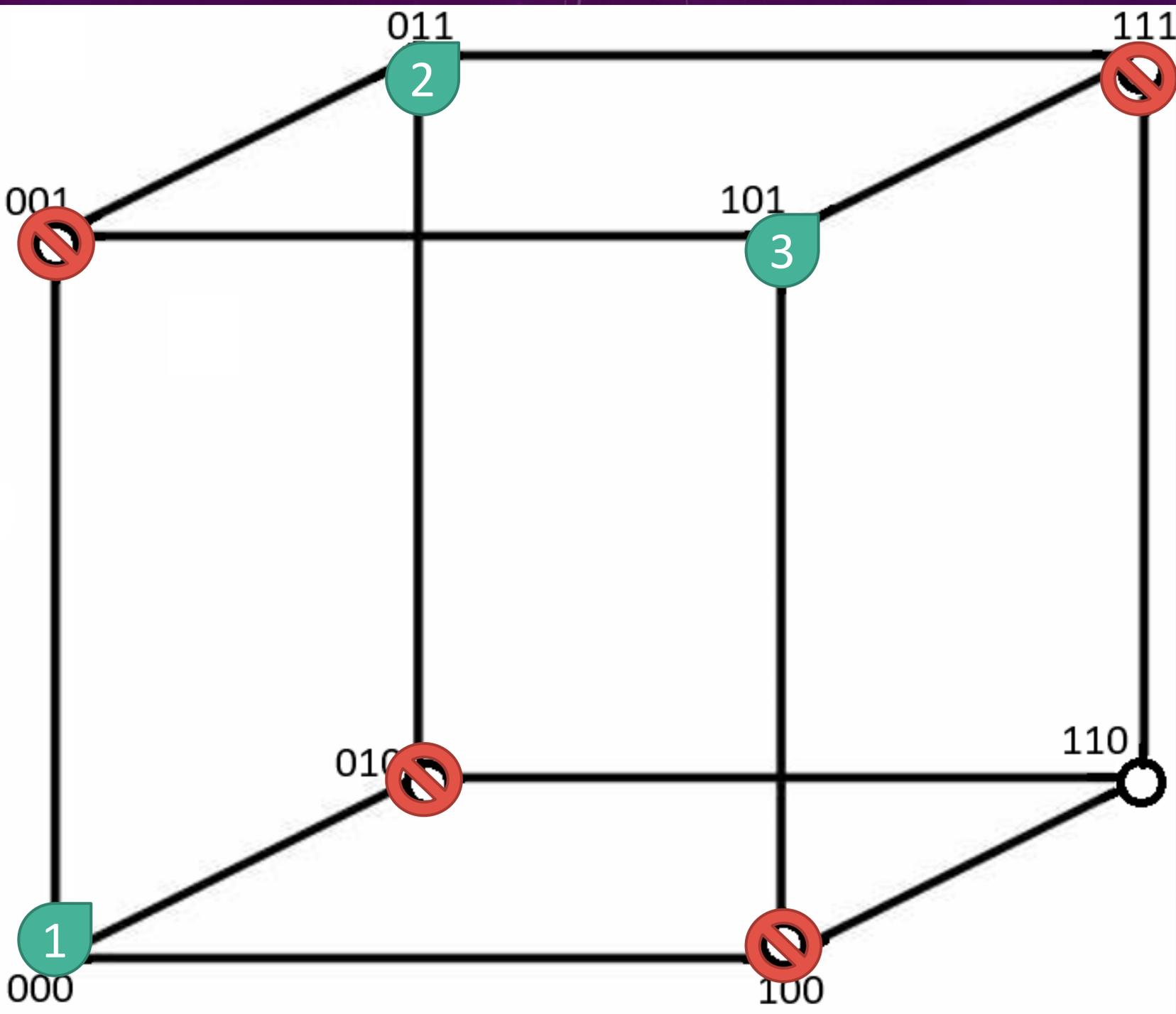
So on, fill the next available channel and reserve each channel within a hamming distance of 1 as un-assigned



HAMMING CUBE

8 POSITIONS

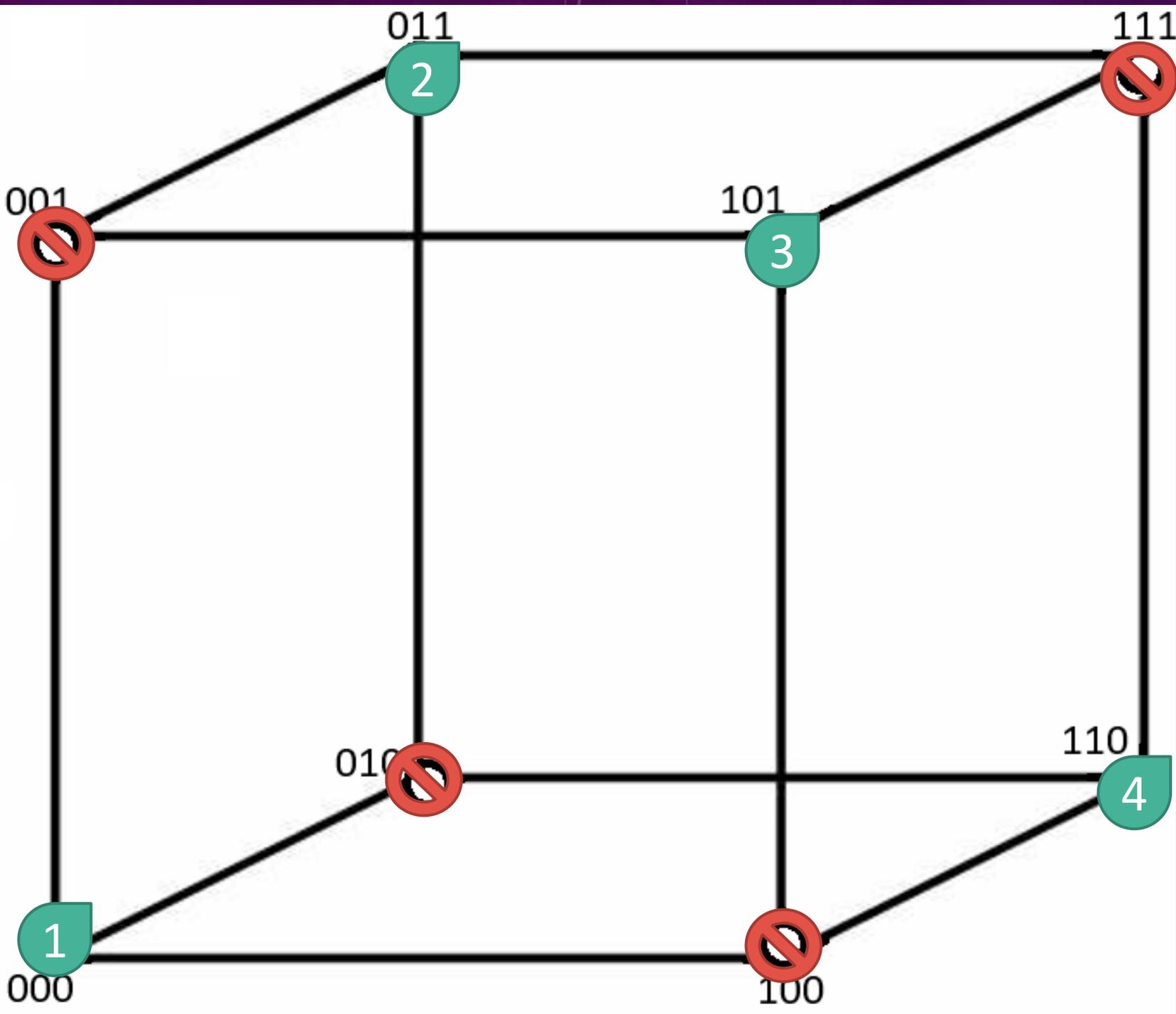
So on, fill the next available channel and reserve each channel within a hamming distance of 1 as un-assigned



HAMMING CUBE

8 POSITIONS

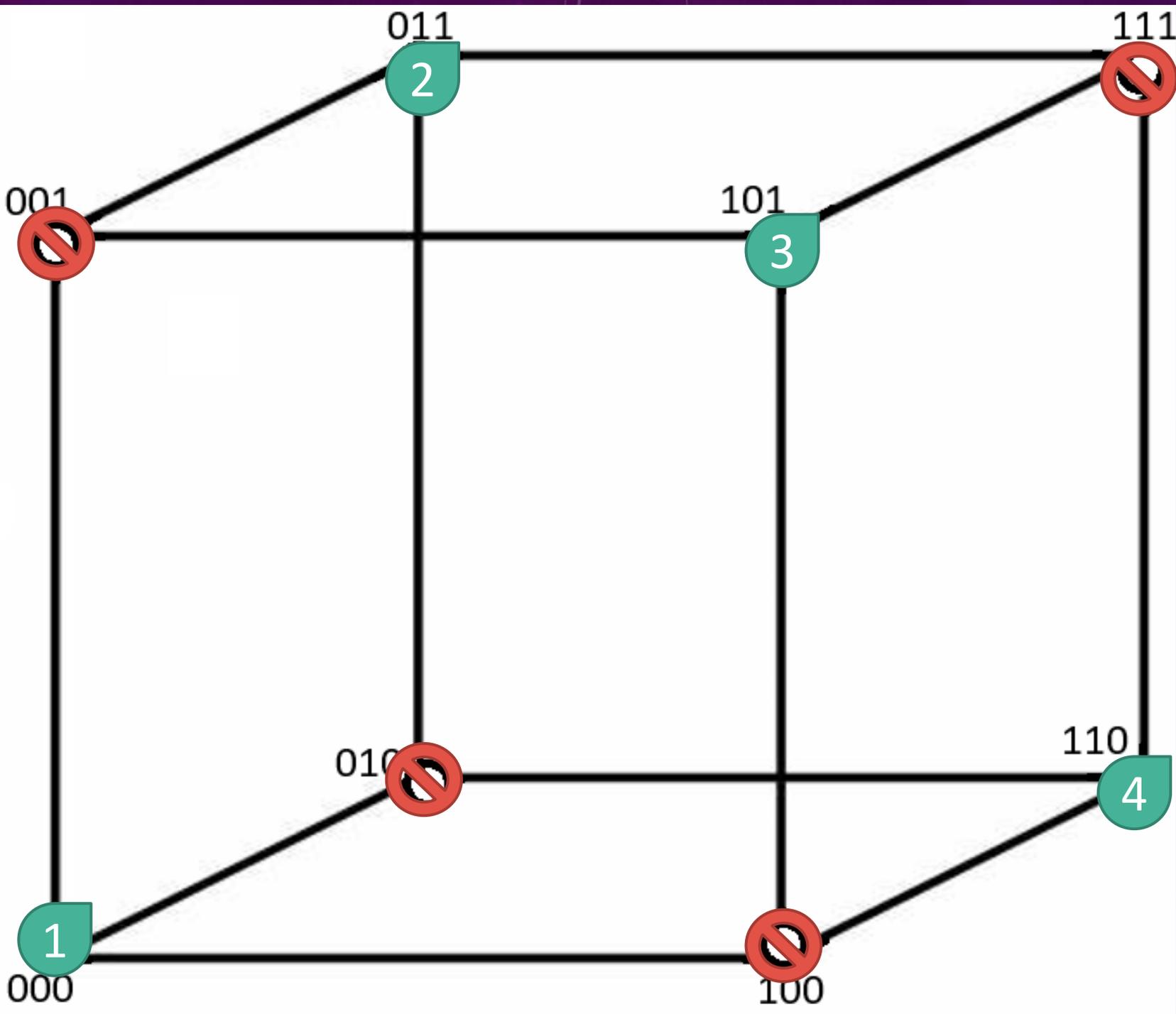
So on, fill the next available channel and reserve each channel within a hamming distance of 1 as un-assigned



HAMMING CUBE

8 POSITIONS

So on, fill the next available channel and reserve each channel within a hamming distance of 1 as un-assigned



HAMMING CUBE

8 POSITIONS

An 8-to-1 MUX can accommodate 4 signals packed in this way.

Multiplexers are commonly produced in 2-to-1, 4-to-1, 8-to-1, and 16-to-1

HAMMING LINE

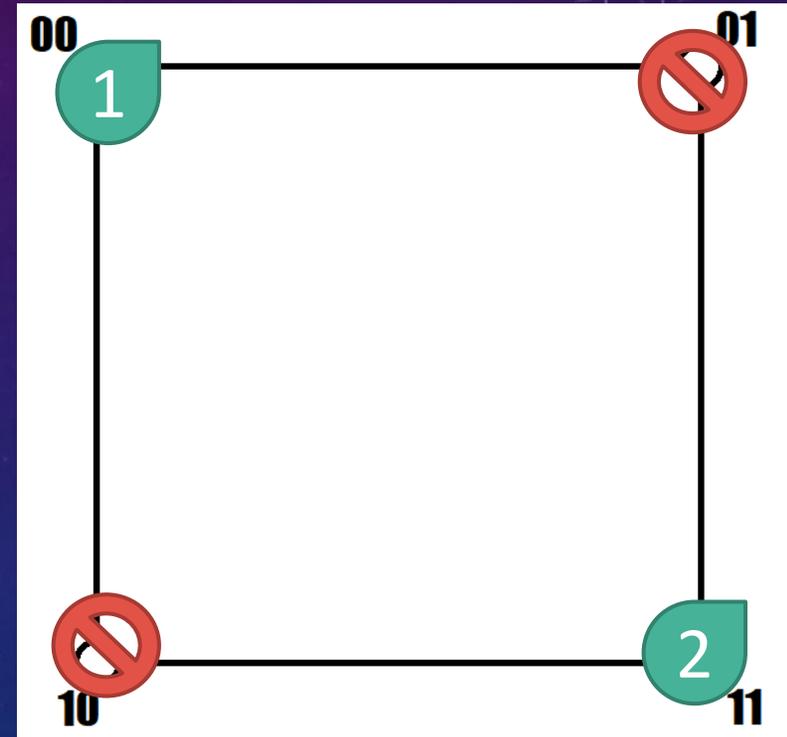
2 POSITIONS



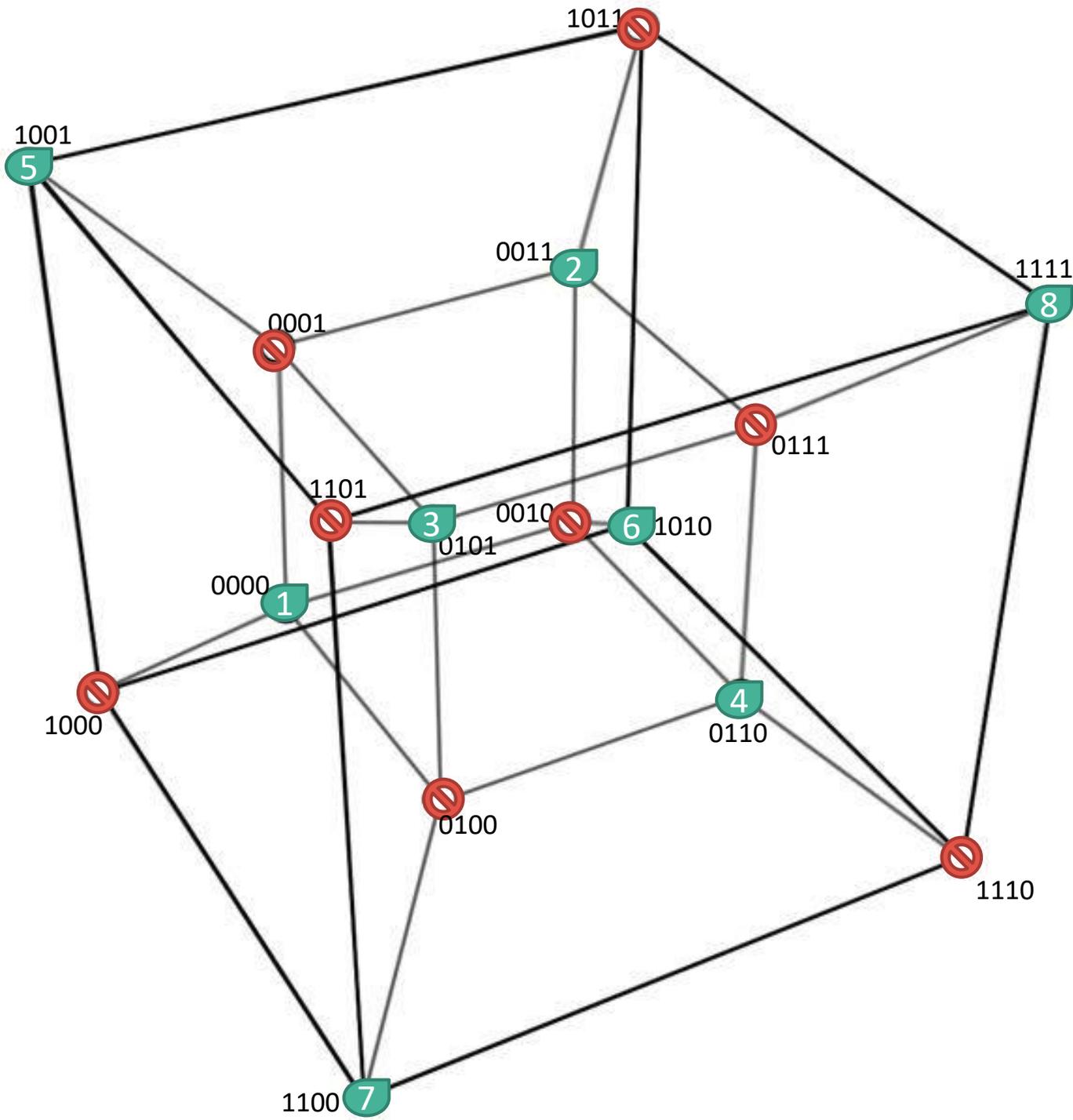
A 2-to-1 MUX (switch) can accommodate 1 signal packed in this way.

HAMMING SQUARE

4 POSITIONS



A 4-to-1 MUX can accommodate 2 signals packed in this way.



THE TESSERACT

16 POSITIONS

A 16-to-1 MUX can accommodate 8 signals packed in this way.



*BEYOND THE LIMITS OF
THE FEEBLE HUMAN MIND
N POSITIONS
 $N \rightarrow \infty$*

A 2^n -to-1 MUX can
accommodate 2^{n-1} signals
packed in this way.

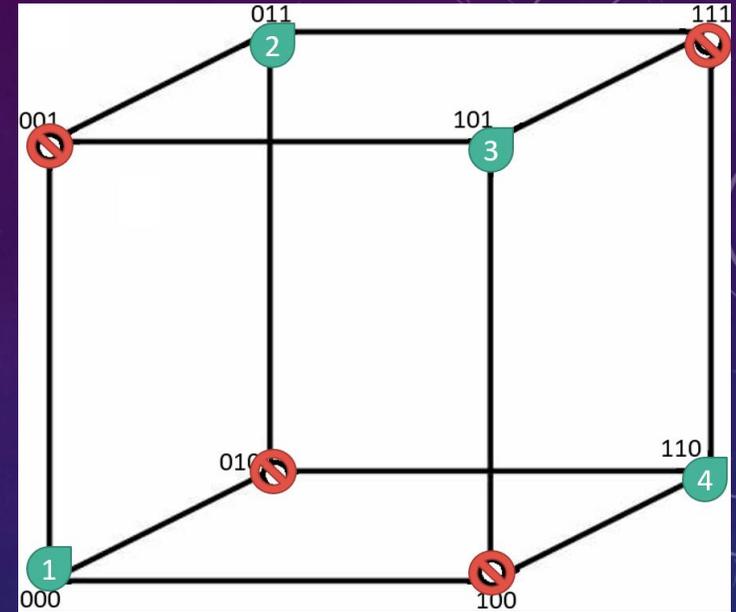
OPTIMAL CHANNELIZATION

Apply this principle to our example:

Based on the Hamming Cube, An 8-to-1 MUX can accommodate 4 signals if each one is to have empty failover channels.

Our MUX has 5 signals to assign. What now?

- 1). Use a bigger MUX (16 to 1)
- 2). Identify another detection method



8 to 1 MUX				
Channel	A2	A1	A0	Sensor
0	0	0	0	Pressure Sensor 1
1	0	0	1	Pressure Sensor 2
2	0	1	0	5V Voltage Reference
3	0	1	1	Temp. Sensor 1
4	1	0	0	Temp. Sensor 2
5	1	0	1	N/A
6	1	1	0	N/A
7	1	1	1	N/A

OPTIMAL CHANNELIZATION

1). Use a bigger MUX (16 to 1)

Depending on your project, this can be a viable solution. A larger MUX adds almost no cost, weight, or complexity. Commonly produced MUX do not differ significantly in reliability between 'sizes'

16 to 1 MUX					
Channel	A3	A2	A1	A0	Sensor
0	0	0	0	0	Pressure Sensor 1
1	0	0	0	1	N/A
2	0	0	1	0	N/A
3	0	0	1	1	Pressure Sensor 2
4	0	1	0	0	N/A
5	0	1	0	1	5V Voltage Reference
6	0	1	1	0	Temp. Sensor 1
7	0	1	1	1	N/A
8	1	0	0	0	N/A
9	1	0	0	1	Temp. Sensor 2
10	1	0	1	0	N/A
11	1	0	1	1	N/A
12	1	1	0	0	N/A
13	1	1	0	1	N/A
14	1	1	1	0	N/A
15	1	1	1	1	N/A

OPTIMAL CHANNELIZATION

2). Identify another detection method

One of these signals is a 5V reference voltage, so we know its expected value. Let's say the nominal range of the pressure and temperature sensors are 10-15 volts – if this is the case, an erroneously telemetered 5V could still be detected as out-of-family, in the same manner as the constant 0V of the unassigned channel.

8 to 1 MUX				
Channel	A2	A1	A0	Sensor
0	0	0	0	Pressure Sensor 1
1	0	0	1	5V Voltage Reference
2	0	1	0	N/A
3	0	1	1	Pressure Sensor 2
4	1	0	0	N/A
5	1	0	1	Temp. Sensor 1
6	1	1	0	Temp. Sensor 2
7	1	1	1	N/A

NOTE ON DETECTION METHODS

In our example, the pressure and temperature sensors share a nominal voltage range, 10-15V. 10V on the pressure channel may mean 20 PSI, while 10V on the temp channel is 70C.

If your signals have sufficiently different expected ranges, you may be able to assign them in Hamming-adjacent channels and still achieve detection.

In this way you achieve high information density and still detect all addressing failures

8 to 1 MUX				
Channel	A2	A1	A0	Sensor
0	0	0	0	Pressure Sensor 1
1	0	0	1	Liquid Level Sensor 1
2	0	1	0	Liquid Level Sensor 2
3	0	1	1	Pressure Sensor 2
4	1	0	0	Liquid Level Sensor 3
5	1	0	1	Pressure Sensor 3
6	1	1	0	Pressure Sensor 4
7	1	1	1	Liquid Level Sensor 4

CURRENT STATE-OF-THE-ART

- Random channelization
- Alphabetical channelization
- Based on hardware physical location

What I have communicated is a best-practice which improves on this state-of-the-art by offering a technique to increase reliability... for free!

SUMMARY

- An addressing error causes the software to request, or the MUX to telemeter, a Hamming-Adjacent channel instead of expected
- If the incorrect data is used in calculations, expect errors
- An un-assigned MUX channel will return a constant 0V
- Make sure addressing errors always cause your sensors to telemeter an out-of-range value, by placing un-assigned channels or sensors with incompatible ranges in Hamming-Adjacent channels
- Develop detection for out-of-range sensor outputs
- Detected errors can be diagnosed, fixed, or prevented from propagation

Contact: Gwyer Sinclair, GWYER.Q.SINCLAIR@NASA.GOV

BACKUP – SENSOR VOLTAGES

Sensors convert some physical phenomena into a voltage. Software reads this voltage and interprets it to generate the reading.

A certain pressure sensor experiencing 20 PSI produces 10V on its output line.

A certain temperature sensor experiencing 50 C produces 8V on its output line.

If the Temp Sensor's measurement is erroneously telemetered, software will think it is seeing 8V from a pressure sensor, and interpret that as 15 PSI. The software will underestimate the pressure.

BACKUP – GRAY CODE

What if my product addresses using Gray Code?

Gray Code is an ordering of Binary expressions such that each successive value is Hamming-Adjacent (differs by only one bit)

Often used in error correction, and to account for latency of electromechanical switches

Gray Code changes the order, not the number of channels
The packing technique described here will still allow you to detect addressing errors

Decimal	Binary Expression				Gray Expression			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

BACKUP – MULTIPLEXER TREES

What if my product needs to select from more than 16 signals on a single data line?

Larger MUX can be created from combinations of smaller MUX. This channelization technique will still be effective at allowing detection of addressing errors, though more complex with added hw.

BACKUP – MULTIPLEXER TREES

The channelization for a 32-to-1 MUX is shown at right. There are 16 channels with no hamming adjacency.

channel	A4	A3	A2	A1	A0
0	0	0	0	0	0 X
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1 X
4	0	0	1	0	0
5	0	0	1	0	1 X
6	0	0	1	1	0 X
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1 X
10	0	1	0	1	0 X
11	0	1	0	1	1
12	0	1	1	0	0 X
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1 X
16	1	0	0	0	0
17	1	0	0	0	1 X
18	1	0	0	1	0 X
19	1	0	0	1	1
20	1	0	1	0	0 X
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1 X
24	1	1	0	0	0 X
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1 X
28	1	1	1	0	0
29	1	1	1	0	1 X
30	1	1	1	1	0 X
31	1	1	1	1	1