PID (Partial Inversion Data):
an M-of-N Level-Encoded Transition Signaling Protocol for Asynchronous Global Communication

Marco Cannizzaro and **Luciano Lavagno**

Dipartimento di Elettronica

Politecnico di Torino, Turin, Italy

Email: {marco.cannizzaro, luciano.lavagno}@polito.it

ASYNC12

May 7-9, 2012

Asynchronous data communication

- Delay-Insensitive (DI) Codes (= unordered codes)
  - Provide timing-robust communication:
    - Tolerant to arbitrary bit skew, P/V/T variability.
- NRZ (2-phase) codes
  - Potential better throughput and less power than RZ (4-phase)
    - no ’spacer code’ is required between any pair of valid codewords.

Level Vs. Transition Encoded

- Level Encoded
  - given a codeword, the encoded data can be directly extracted by using a combinational logic function
    - any codeword corresponds to one and only one symbol
- Transition Encoded
  - the encoder and decoder need to store at least one past codeword.

<table>
<tr><th>Level Encoded</th><th>Transition Encoded</th></tr>
<tr><td>1-of-2 LEDR</td><td rowspan="2">M-of-N Transition Encoded</td></tr>
<tr><td>1-of-N LETS</td></tr>
</table>

DI 2-phase Background: 1-of-2 LEDR

- ***2 wires per bit***
- ***Level-encoding***
  - Data rail: holds actual data value
  - Parity rail: holds parity value
- ***Alternating-phase protocol***
  - Encoding parity alternates between odd and even

Bit value

Phase	0	1
Even	0 0	1 1
Odd	0 1	1 0

Data Rail

Parity Rail

DI 2-phase Background:
1-of-4 LETS

- ***4 wires per 2 data bits***
- ***Alternating-phase protocol***
  - 2 codewords for each symbol in each phase

phase	symbol	codeword
ODD	S0	1000/0111
	S1	0100/1011
	S2	0010/1101
	S3	0001/1110
EVEN	S0	1111/0000
	S1	0011/1100
	S2	0101/1010
	S3	0110/1001

DI 2-phase Background:
M-of-N Transition Encoded

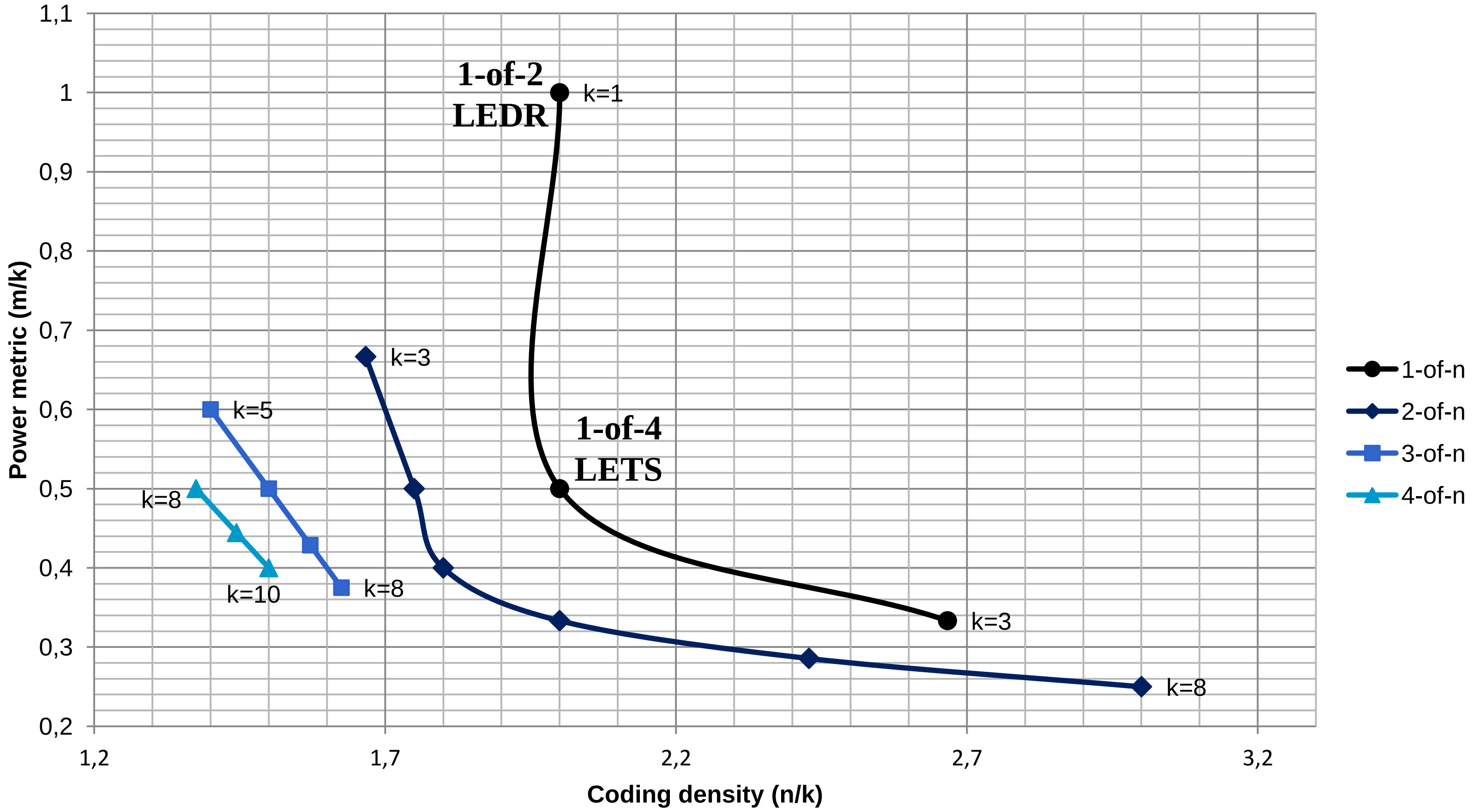
- *m*: number of transitions per transaction
- *n*: number of bits of the codeword
- *k*: max. number of bits encoded

→ Any combination of *m* transitions in the codeword encodes exactly one symbol

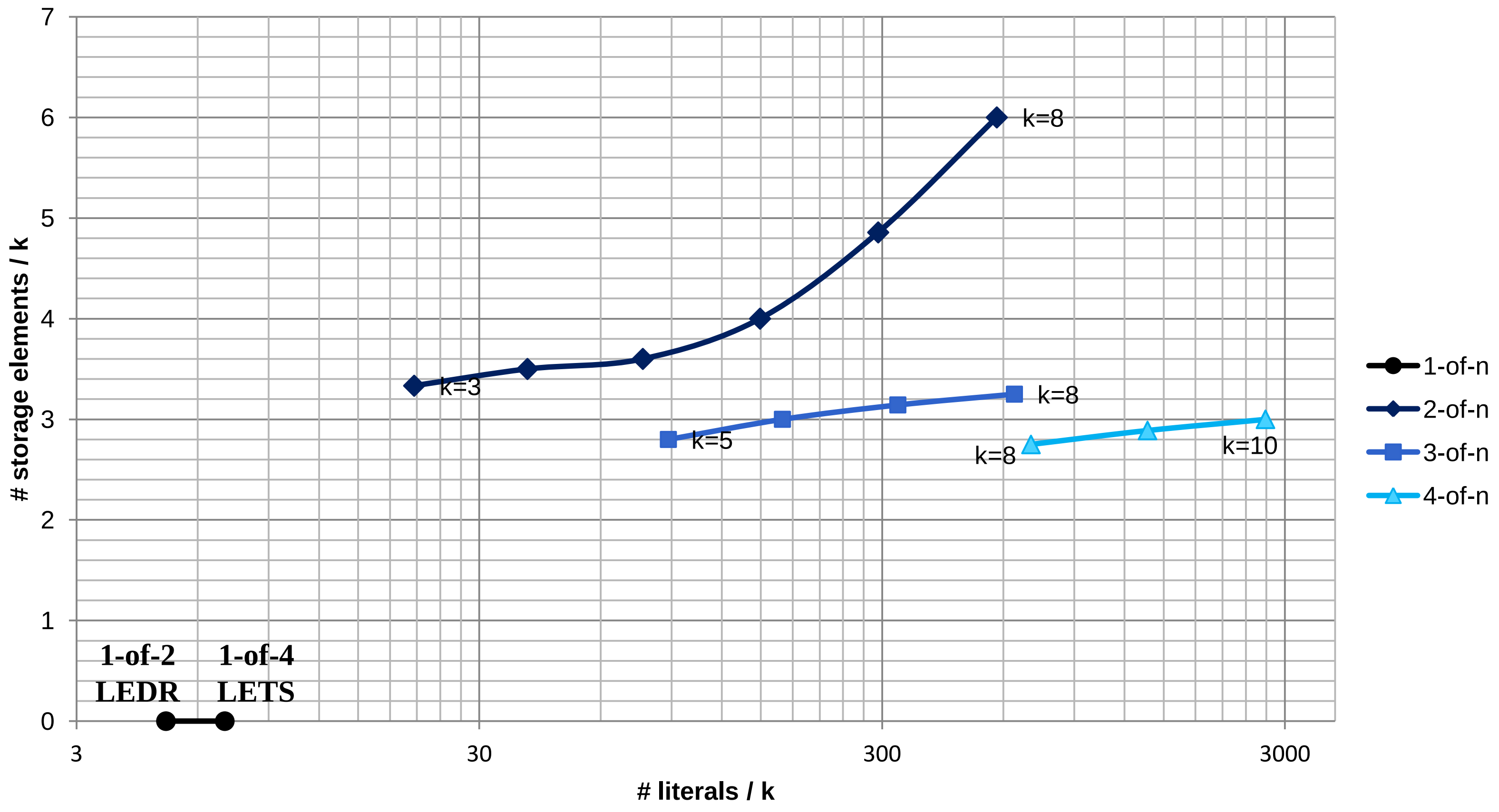
$$\#symbols = \frac{n!}{m!(n-m)!}$$

$$k = floor(\log_2 \#symbols)$$

Comparison:
Power Vs. Coding Density



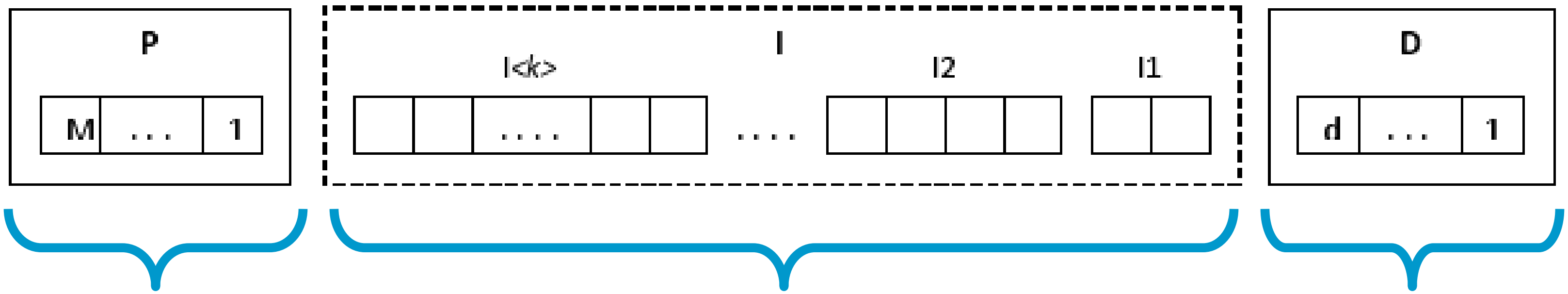
Comparison:
Hardware (decoder) cost



Contribution

Evaluation metric	M-of-N Transition Encoded	1-of-2 LEDR 1-of-N LETS	M-of-N PID
Coding efficiency	good	bad	good
Power consumption	good	bad	good
Hardware cost	bad	good	good

PID codeword



The **Parity** field (P) always ensures M transitions in the codeword.

The **Inversion** field (I) carries two pieces of information:

1. whether the data field (D) is inverted or not and
2. the value of k encoded bits which are not in the data field (D).

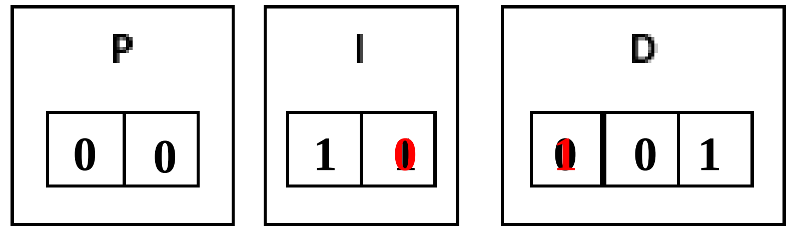
The **Data** field (D) carries the value of the first d bits of the encoded data (inverted or not according to the inversion field).

PID: the idea

- By optionally inverting all the bits of the data field (D) we reduce the maximum number of transitions to the floor of *d/2* for any transaction.

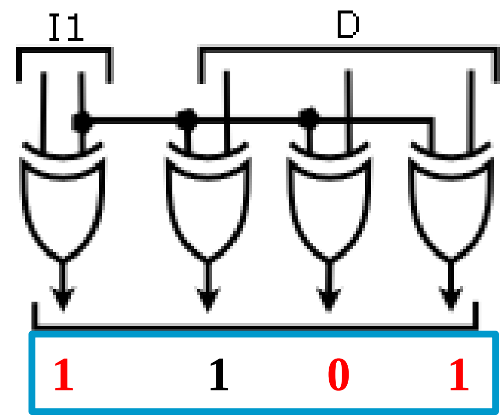
- The inversion field (I) (which always has 0 or 1 transitions) is composed of sub-fields I<x> and each of them corresponds to one encoded data bit.
  This increases the number of encoded data bits without increasing the number of transitions M in the codeword (i.e. improves the power efficiency of the code).

Example: the 2-of-7 PID code



Last codeword: 00.11.001
- ➢Encoded data: 0110

Next data to be encoded: 1101
- ➢Next codeword: 00.10.101

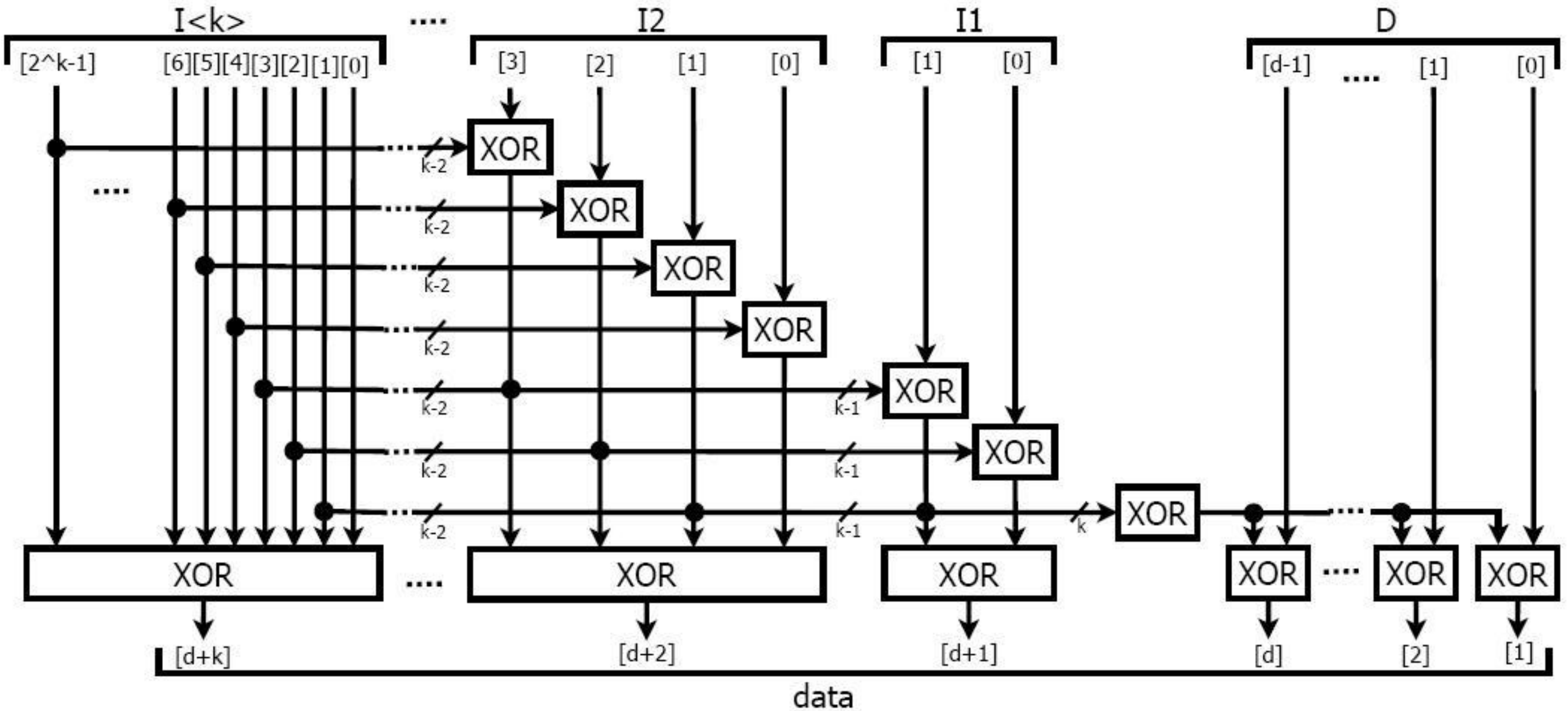


M-of-N PID codes

# data bits	Proposed M-of-N codes						
	d=1	d=2	d=3	d=4	d=5	d=6	d=7
1	1-of-2	-	-	-	-	-	-
2	1-of-4	2-of-4	-	-	-	-	-
3	1-of-8	2-of-6	-	-	-	-	-
4	1-of-16	2-of-10	2-of-7	-	-	-	-
5	1-of-32	2-of-18	2-of-11	3-of-9	-	-	-
6	1-of-64	2-of-34	2-of-19	3-of-13	3-of-10	-	-
7	1-of-128	2-of-66	2-of-35	3-of-21	3-of-14	4-of-12	-
8	1-of-256	2-of-130	2-of-67	3-of-37	3-of-22	4-of-16	4-of-13
9	1-of-512	2-of-258	2-of-131	3-of-69	3-of-38	4-of-24	4-of-17

Codes in grey are not Pareto-optimal and can be replaced with other codes which have better coding efficiency.

M-of-N PID decoder HW



The hardware for a particular M-of-N1 code can be reused for any M-of-N2 code where N2 < N1, if the extra inputs in the inversion field are not used.

M-of-N PID Encoding algorithm

Step 1: The Hamming distance is computed between the first d bits of the new data and the data field (D) of the previous codeword.

Step 2: Each one of the other k data bits is compared to the corresponding bit of the previous data and the index of the inversion sub-field that must have a transition is selected.

Step 3.1: If one inversion field must be flipped, the algorithm:

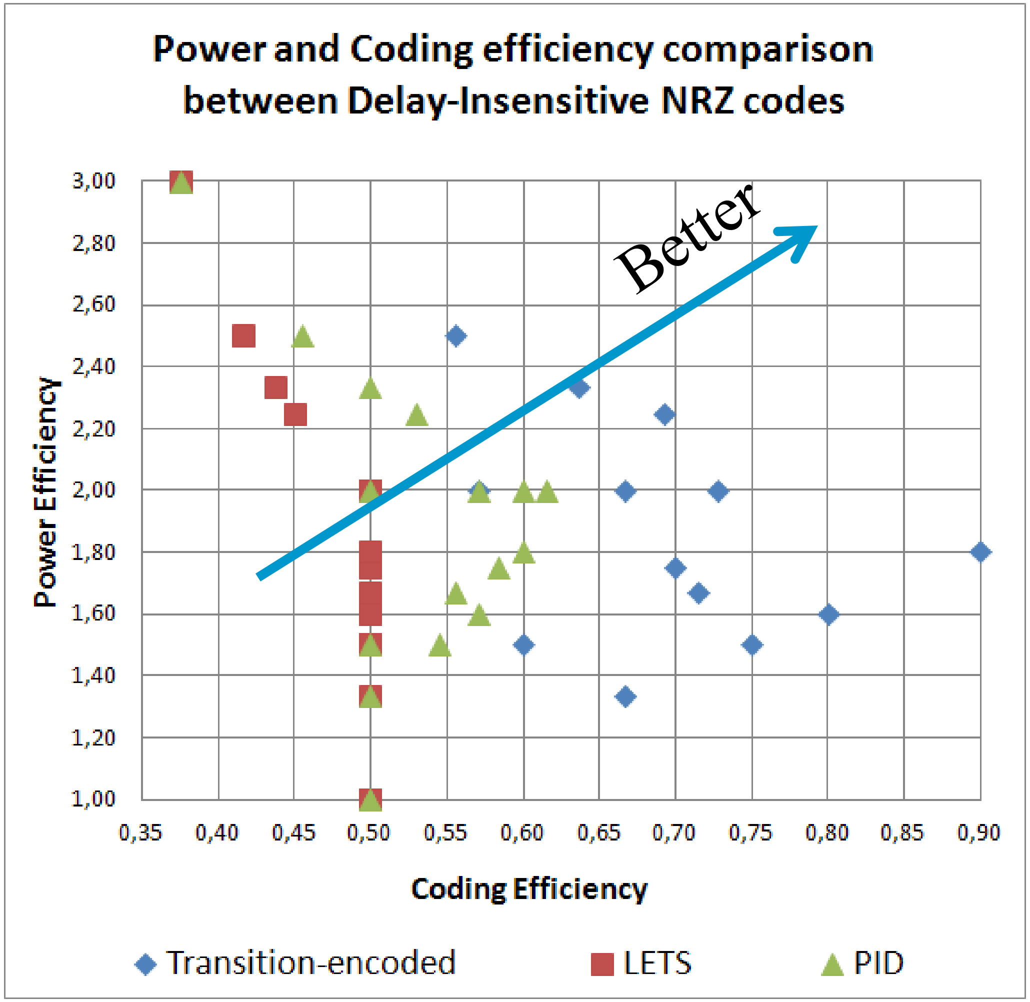
1. Checks whether the data will be inverted in the new data field or not.
2. Looks for and flips the bit within the inversion sub-field.

Step 3.2: If none inversion field must be flipped, the algorithm only checks whether the data will be inverted in the new data field or not.

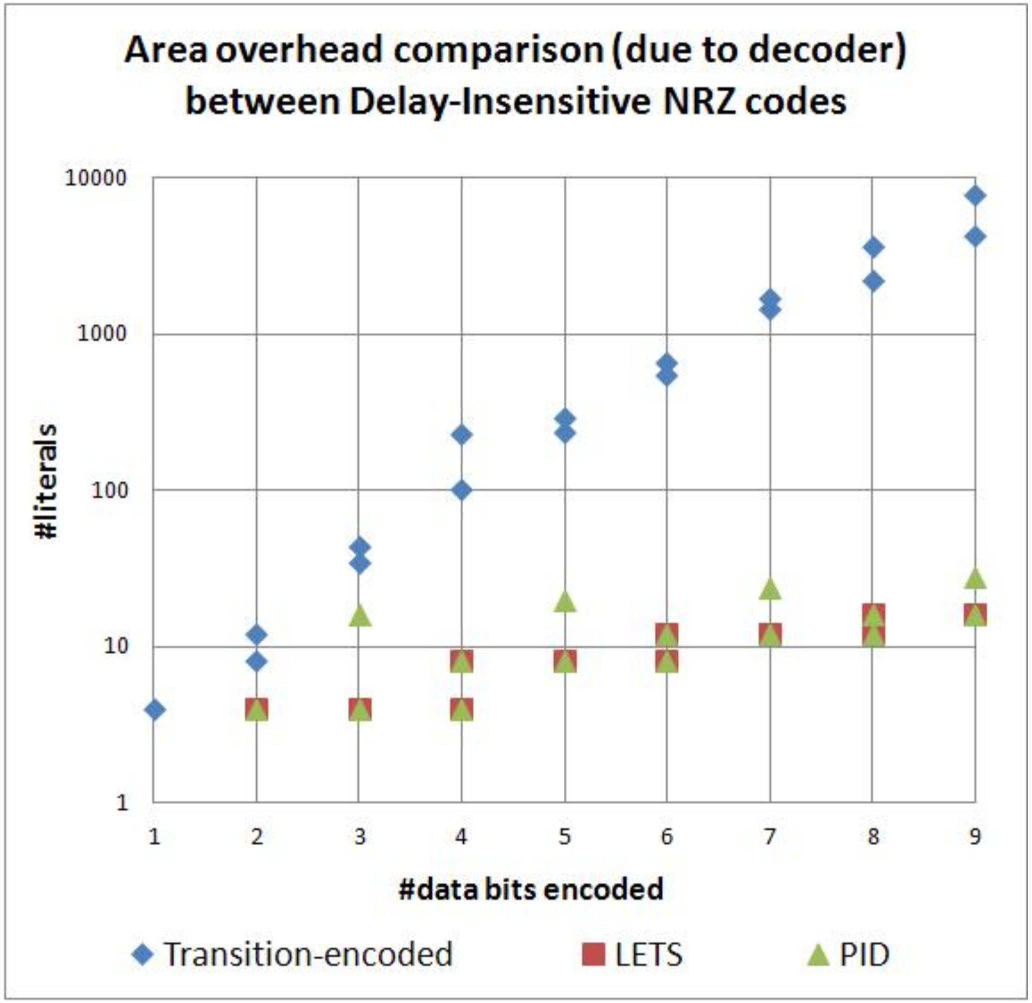
Step 4: Data is inverted or not to generate the data field (D).

Step 5: Between 0 and M bits of the parity field are flipped in order to always have M transitions in the codeword.

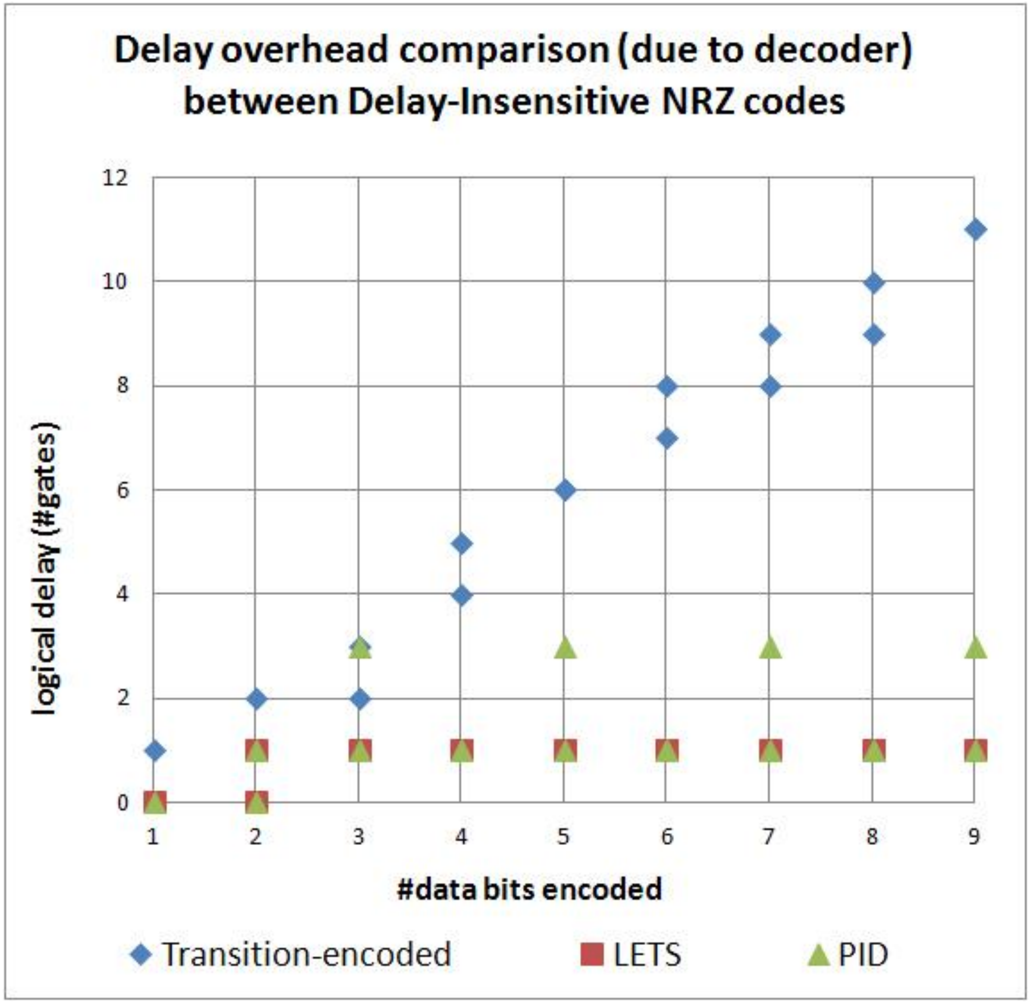
Results: Power and Coding efficiency



Results: Area overhead (due to decoder)



Results: Delay overhead (due to decoder)



Conclusion: the PID code…

- …is a **Delay-Insensitive M-of-N** protocol, where only M wires flip for each data transaction.
- …is a **NRZ** code, having significant power and throughput benefits with respect to Return-to-Zero (RZ) codes.
- …is **Level-encoded**, meaning that the decoding process simply uses the values of the codeword.
- …has a **generic encoding algorithm and decoder implementation** (that works for any M-of-N PID code).

Conclusion: PID results

PID comparison	Coding Efficiency	HW overhead
1-of-N LETS	better/equal	equal (but LETS has no generalization)
M-of-N Transition Encoded	worse/equal	better

In particular, the **2-of-7 PID code**, which encodes 4 data bits in 7 codeword wires, **Pareto dominates all other DI NRZ codes**.

Thank you for your attention!

Marco Cannizzaro

Dipartimento di Elettronica

Politecnico di Torino, Turin, Italy

Email: marco.cannizzaro@polito.it