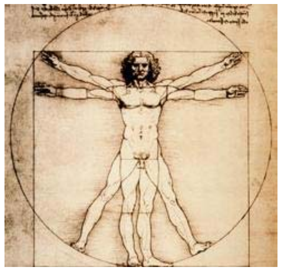
Hardware Optimization of error control decoders

Emmanuel Boutillon

CNRS, UMR 3192, Lab-STICC

Centre de Recherche - BP 92116

F-56321 Lorient Cedex - FRANCE

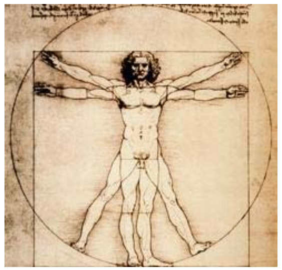


International Workshop on Channel Coding and Cooperation in Wireless Communications

This work has been donne in the framework of DaVinci project (FP7).

With the contributions of :

- **Laura Conde-Canencia**
- **Ashwani Singh**
- **Ali Al-Ghouwayel**
- **Pierre Bomel**
- **Yang-Yang Tang**
- **Cedric Marchand**



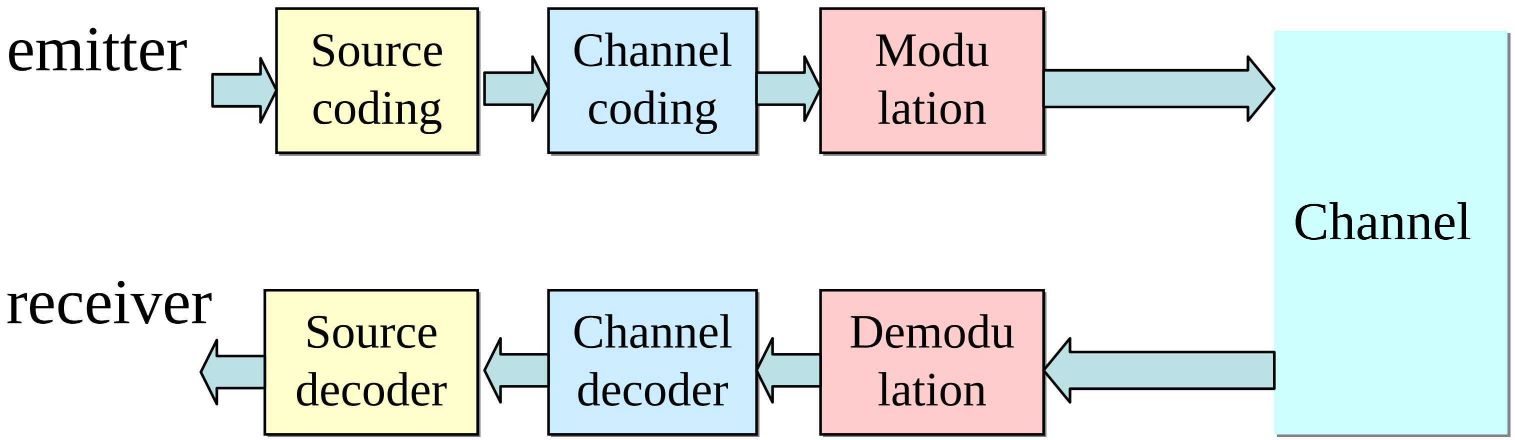
International Workshop on Channel Coding and Cooperation in Wireless Communications

OUTLINE

- MOTIVATION
- REDUCED MONTE CARLO SIMULATION
- ENTROPY INSPIRED DISTANCE
- HARDWARE EMULATION
- CONCLUSION

Motivation

Design of a communication system...

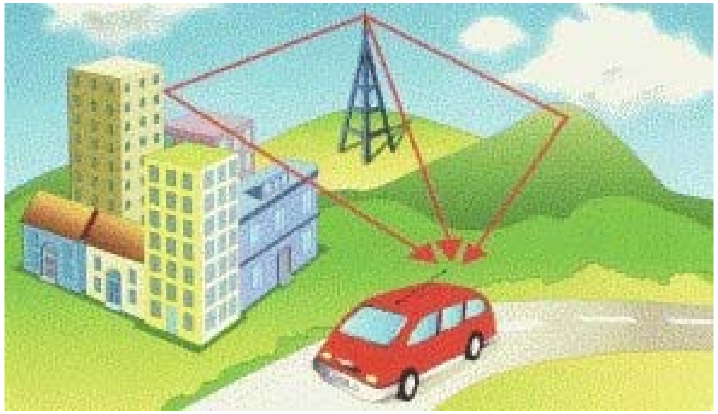


...find the best complexity-performance trade-off

Motivation

Performance:

- BER
- Jammer rejection
- time of synchronization…
- …



Complexity:

- area, power dissipation
- time to market
- …

algorithm
ADC resolution,
sampling frequency,
fixed precision

A very complex problem...

Monte-carlo simulation

- Formal expression of the BER: refer to Proakis
- In practice, estimation of the BER using Monte-carlo simulation
  - Software model of emitter, channel, receiver
  - Emulation of the transmission of *N* bits
  - Estimation of the BER as *Nb_errors*/*N*

VERY EFFICIENT…

…but

TIME CONSUMMING: BER of 10^{-6} *(+-3%) requires* 10^{9} *bits.*

Two separate problems

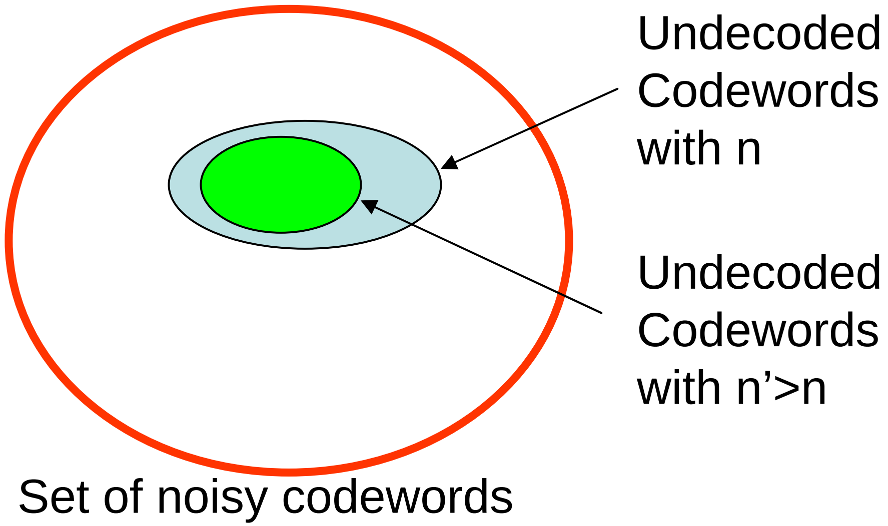
- 1) The code design problem.
  - Theoretical bound
  - Weight estimation using impulse method and its derivative
  - “Smart” MC simulation (semi-random sample)
  - Estimation of the APP distribution using a reduced set of samples => deduce BER
- 2) Evaluate performance of “hardware decoder” vs “ideal decoder”

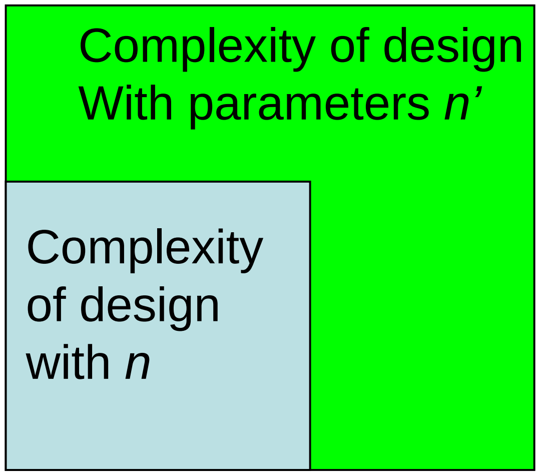
OUTLINE

- **MOTIVATION**
- **REDUCED MONTE CARLO SIMULATION**
- **ENTROPY INSPIRED DISTANCE**
- **HARDWARE EMULATION**
- **CONCLUSION**

REDUCED MONTE CARLO SIMULATION

- **Let us assume that we want to find a best trade-off of a algorithm parameter *n*. assuming that:**
- **1) Decoder complexity is an increasing function of *n*.**
- **2) Performance is a strictly increasing function, i.e. if a noisy codeword is decoded with parameter *n*, than the codeword is decoded with parameter *n'* > *n*.**





In that case, RMCS can be efficient

REDUCED MONTE CARLO SIMULATION

- **Step 1: Run Monte-Carlo simulation with the "worst-case" configuration. Store the undecoded codeword in set *S* ;**
- **Step 2: Re-run decoder with better parameters only in subset *S* to evaluate performances ;**
- **Step 3: perform a classical MCS in order to verify the validity of hypotheses 1 for some configurations.**

RESULT ON DAVINCI CODE (LDPC GF(64))

10^7 frames
(BER 5.10^{-3})

	it =8	*it* =10	*it* =12	*it* =14	*it* =16
***n* =8**	49693	13535	5404	2993	2105
***n* =10**	12340	2569	815	414	291
***n* =12**	4763	893	270	109	62
***n* =14**	2419	467	163	72	42
***n* =16**	1349	223	~~56~~ **71**	22	~~13~~ **22**
***n* =18**	835	151	25	10	4
***n* =20**	592	116	17	7	6
***n* =22**	455	103	14	4	3
***n* =24**	373	86	~~14~~ **21**	5	~~4~~ **8**
***n* =26**	321	91	16	4	3
***n* =28**	262	54	9	5	3
***n* =30**	235	52	8	4	3
***n* =32**	228	52	~~7~~ **19**	4	~~2~~ **8**

Save 3 months of simulation…

OUTLINE

- **MOTIVATION**
- **REDUCED MONTE CARLO SIMULATION**
- **ENTROPY INSPIRED DISTANCE**
- **HARDWARE EMULATION**
- **CONCLUSION**

ENTROPY INSPIRED DISTANCE

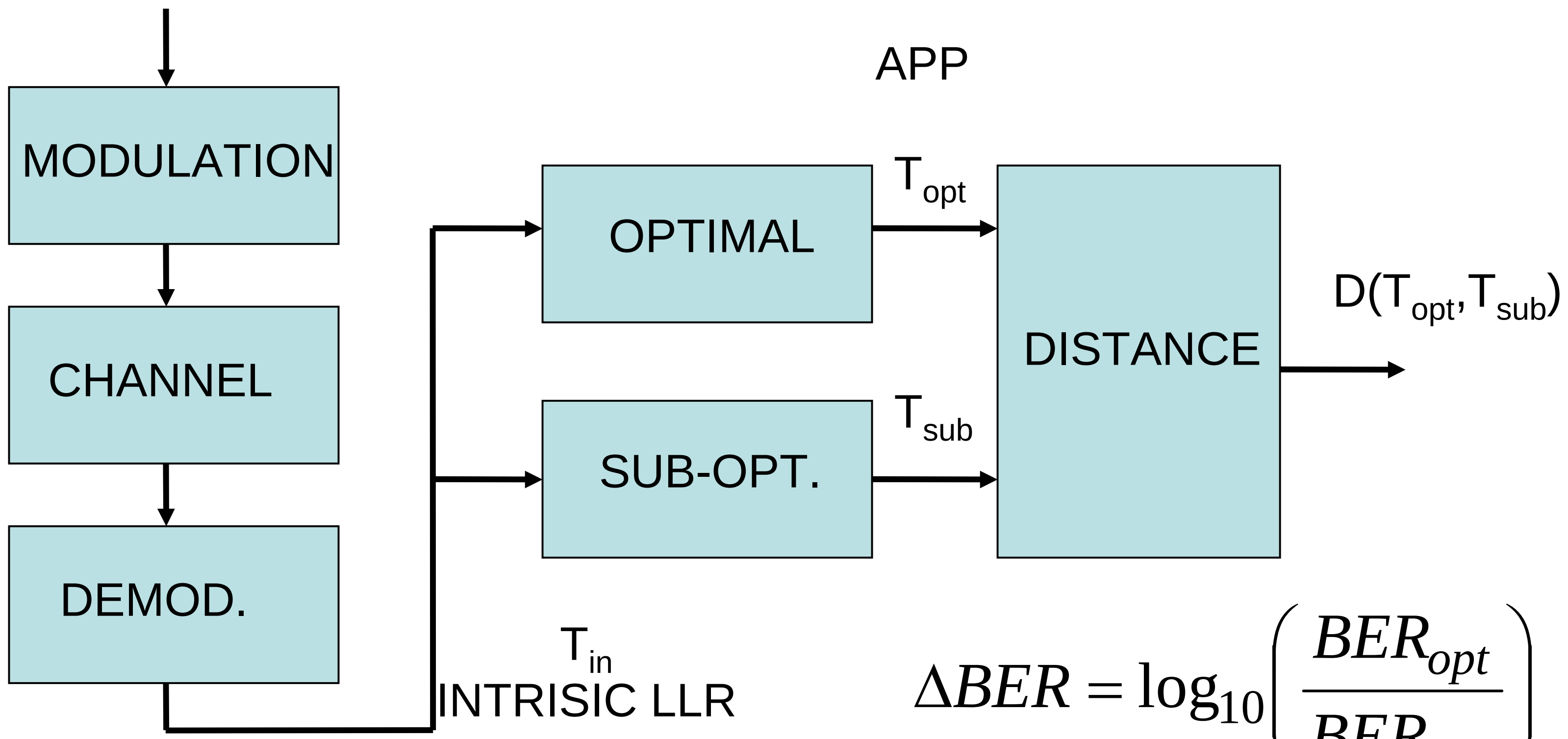
- **CONSTAT : FROM APP TO BER, A HUGE QUANTITY OF INFORMATION IS SUPPRESSED.**

QUESTION 1 : CAN WE EXPLOIT THIS INFORMATION ?

QUESTION 2 : HOW ?

DISTANCE BETWEEN OPTIMAL/SUB-OPT.

SET OF *M* CODEWORDS OF SIZE P (*N*=*M*x*P* *Q*-ary symbols)



$$\Delta BER = \log_{10}\left(\frac{BER_{opt}}{BER_{sub}}\right)$$

FIND D(Topt,Tsub) RELATED TO ΔBER

QUESTION : WHAT DISTANCE TO CHOOSE…

- **LET** $(p_k^0, p_k^1, \ldots, p_k^{Q-1})$ **AND** $(\tilde{p}_k^0, \tilde{p}_k^1, \ldots, \tilde{p}_k^{Q-1})$ **BE THE APP OF THE *k*th DECODED SYMBOL FOR OPTIMAL AND SUB-OPTIMAL DECODERS.**

$$\sum_{i=0}^{Q-1} p_k^i = 1 \qquad \sum_{i=0}^{Q-1} \tilde{p}_k^i = 1$$

- FIRST ATTENT : L1-NORM

$$L_1 = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} \left| p_k^i - \tilde{p}_k^i \right| \right)$$

EXPERIMENTAL RESULT: NO CLEAR CORRELATION BETWEEN L1-log(Tsub, Topt) AND ΔBER.

PROPOSED EXPLANATION: $p = 10^{-6}$ AND $p = 10^{-12}$ ARE REALLY DIFFERENT FROM A DECODING PERSPECTIVE

QUESTION : WHAT DISTANCE TO CHOOSE…

- **LET** $(p_k^0, p_k^1, \ldots, p_k^{Q-1})$ **AND** $(\tilde{p}_k^0, \tilde{p}_k^1, \ldots, \tilde{p}_k^{Q-1})$ **BE THE APP OF THE *k*th DECODED SYMBOL FOR OPTIMAL AND SUB-OPTIMAL DECODERS.**

$$\sum_{i=0}^{Q-1} p_k^i = 1 \qquad \sum_{i=0}^{Q-1} \tilde{p}_k^i = 1$$

- SECOND ATTENT : L1-log MORM $$L_1 = \sum_{k=0}^{N-1}\left(\sum_{i=0}^{Q-1} -\log\left|p_k^i - \tilde{p}_k^i\right|\right)$$

EXPERIMENTAL RESULT: NO CLEAR CORRELATION BETWEEN L1-log(Tsub, Topt) AND ΔBER.

PROPOSED EXPLANATION: *p* =0.7 AND *p*=0.9 ARE REALLY DIFFERENT FROM A DECODING PERSPECTIVE AND -log(0.2) << -log(10^{-5})

QUESTION : WHAT DISTANCE TO CHOOSE…

- **LET** $(p_k^0, p_k^1, \ldots, p_k^{Q-1})$ **AND** $(\tilde{p}_k^0, \tilde{p}_k^1, \ldots, \tilde{p}_k^{Q-1})$ **BE THE APP OF THE *k*th DECODED SYMBOL FOR OPTIMAL AND SUB-OPTIMAL DECODERS.**

$$\sum_{i=0}^{Q-1} p_k^i = 1 \qquad\qquad \sum_{i=0}^{Q-1} \tilde{p}_k^i = 1$$

- THIRD ATTENT : KULLBACK-LEIBLER DIVERGENCE

$$KL = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} - p_k^i \log(p_k^i / \tilde{p}_k^i) \right)$$

EXPERIMENTAL RESULT: NO CLEAR CORRELATION BETWEEN KL(Tsub, Topt) AND ΔBER.

PROPOSED EXPLANATION: *NO EXPLANATION…*

ENTROPY INSPIRED DISTANCE (EID)

- **ENTROPY OF A DISTRIBUTION :** $H(X) = \sum_{i=0}^{Q-1} - p_k^i \log(p_k^i)$

- PROPERTIES : H(X) ≥ 0, H(X)=log(Q) for $p^i=1/Q$, $i=0..Q-1$.

$$EID = \sum_{k=0}^{N-1} \left(\sum_{i=0}^{Q-1} - \left| p_k^i - \tilde{p}_k^i \right| \log\left(\left| p_k^i - \tilde{p}_k^i \right| \right) \right)$$

EXPERIMENTAL RESULT: USEFUL TOOLS TO PREDICT THE ΔBER !!!

PROPERTIES OF EID

- **IF** $Arg\max(p^i) = Arg\max(\tilde{p}^i)$

THEN EID IS A DISTANCE, I.E.

EID(X,X')=0 => X=X' (note that EID([0,0,0,1], [1,0,0,0]) = 0)

EID(X,X') = EID(X',X)

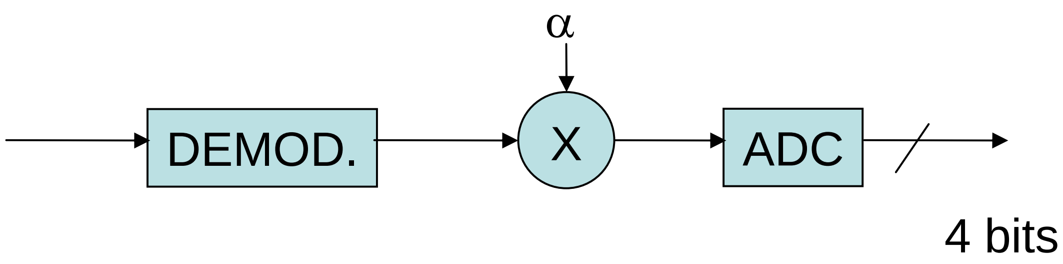
EID(X,Y) + EID(Y,Z) $\geq$ EID(X,Y)

AT THE MOMENT, PROVEN FOR Q=2 UP TO 12.

…STILL NEED A PROOF FOR Q > 12

SIMULATION RESULT : CASE 1

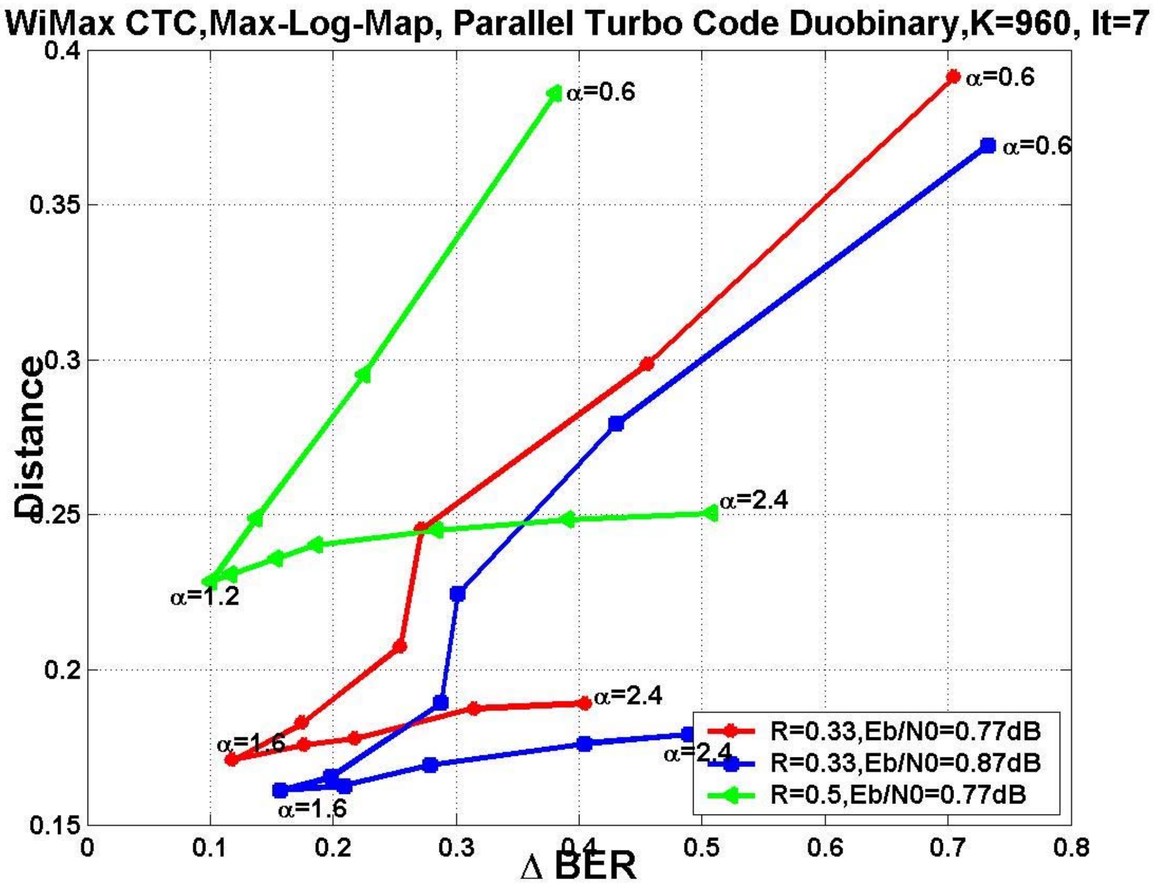
- **DUO BINARY TURBO-CODE**
- **SEARCH FOR THE OPTIMAL QUANTIZATION FACTOR**



- **IF α too small => ERASURE**
- **IF α too big => SATURATION (HARD DECODER)**

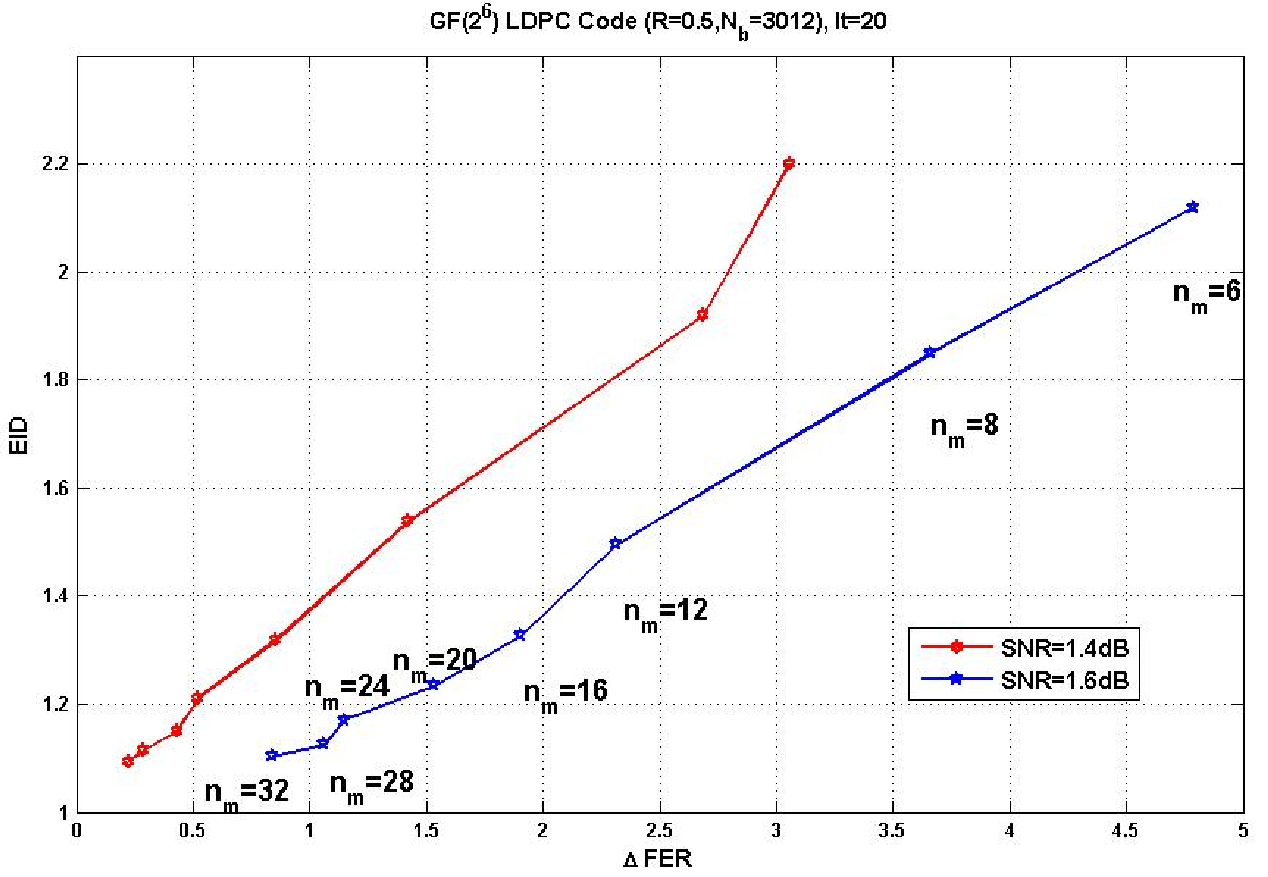
THERE IS AN OPTIMAL VALUE OF α

SIMULATION RESULT : CASE 1



SIMULATION RESULT : CASE 2

- NON BINARY LDPC DECODER ON GF(64) USING EMS ALGORITM



OUTLINE

- **MOTIVATION**
- **REDUCED MONTE CARLO SIMULATION**
- **ENTROPY INSPIRED DISTANCE**
- **HARDWARE EMULATION**
- **CONCLUSION**

Software simulation

Three methods to reduce the simulation time:

a) code optimization

b) powerful computing

c) parallel computing
(One Mbps for a turbo-decoder with a cluster of 16 PCs)

also use hardware emulation

Current methodology

Software

Algorithm

C programs

Compilation

Validation/optimization with long simulations

Fix specifications

Hardware

VHDL programs

Synthesis, place and route operations

Validation

Final prototype

Proposed methodology

Software

Algorithm

C programs

Compilation

Validation/optimization

Fix algorithm + Set of non-specified parameters

Hardware

Generic VHDL programs, IP

Synthesis, place and route operations (on FPGA)

Hardware simulation/validation

Final prototype

Channel emulation

Type of communication channel:

- AWGN
- Rice
- Rayleigh

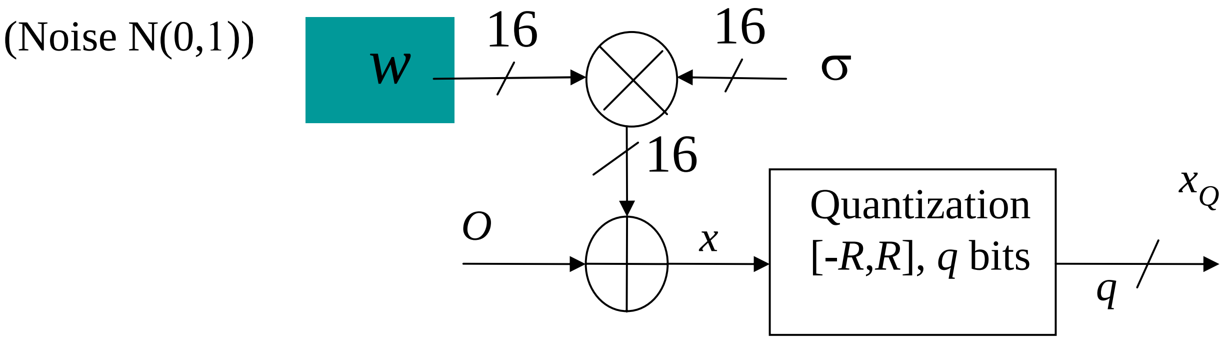
...

All those channels can be derived from Gaussian Noise (with ARMA filter, non-linear operators).

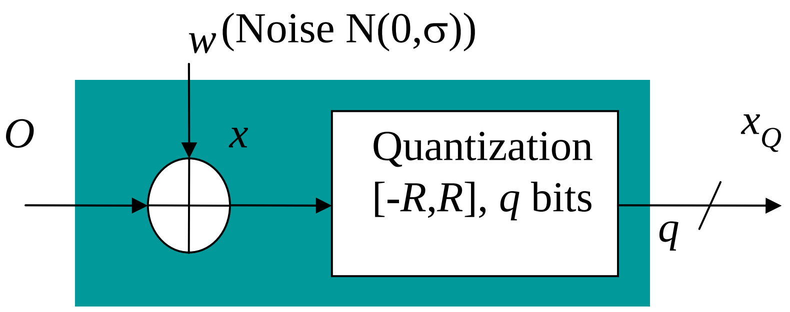
=> Need a White Gaussian Noise Generator (WGNG)

Nouvelle version

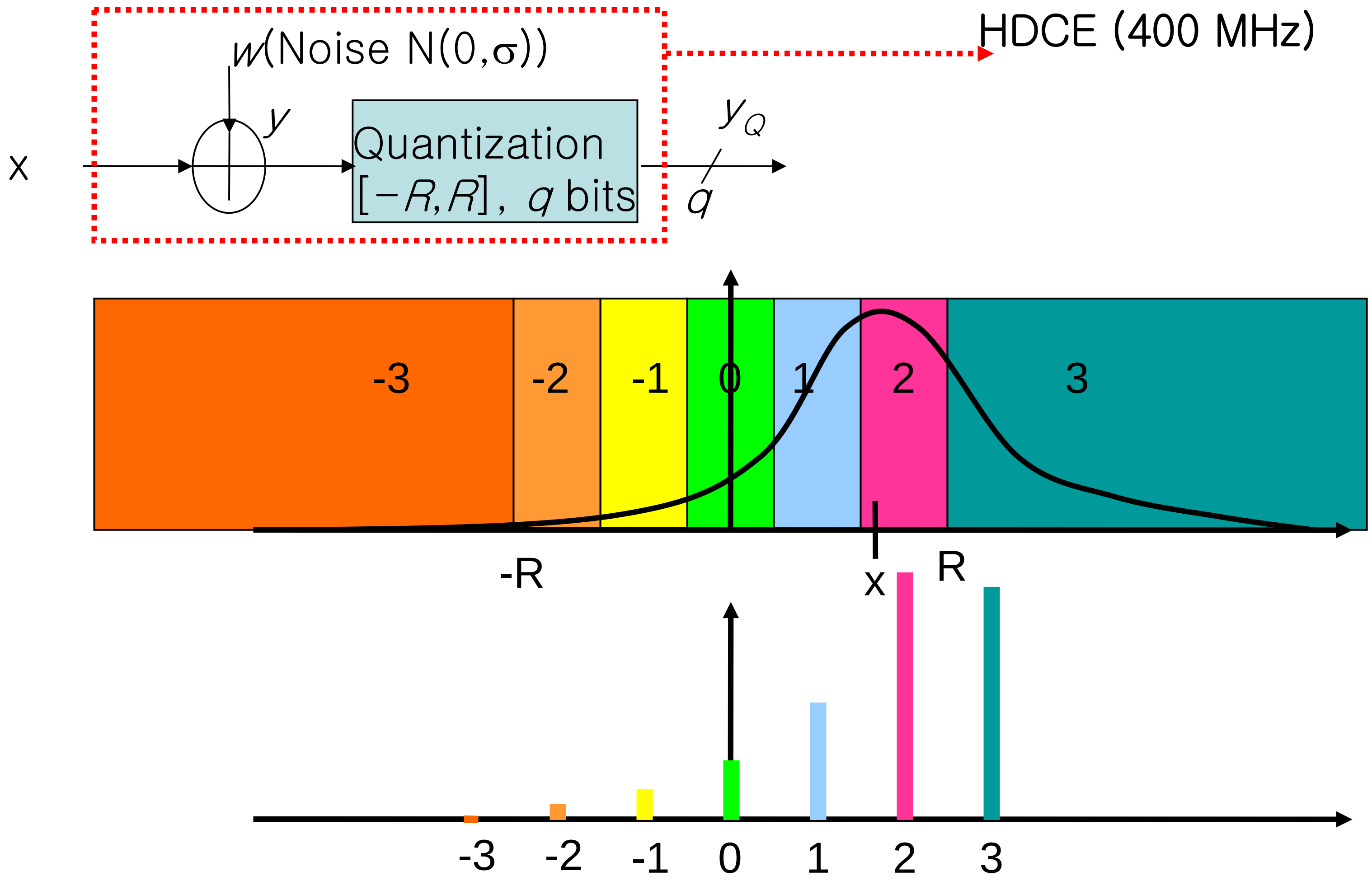
- **Old approach**



- **New approach: emulation of Channel+quantization**



DISCRETE HARDWARE CHANNEL EMULATOR

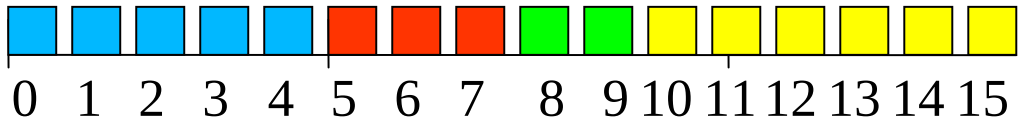


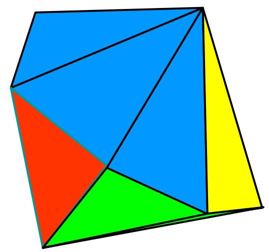
- FOR A GIVEN x, σ, R, EQUIVALENT TO A DISCRETE R.V.

29

Naive method to generate a discrete variable

- **Throw a 16 face dice => it gives a color**





P(x=B)=5/16
P(x=R)=3/16
P(x=G)= 2/16
P(x=W)= 6/16

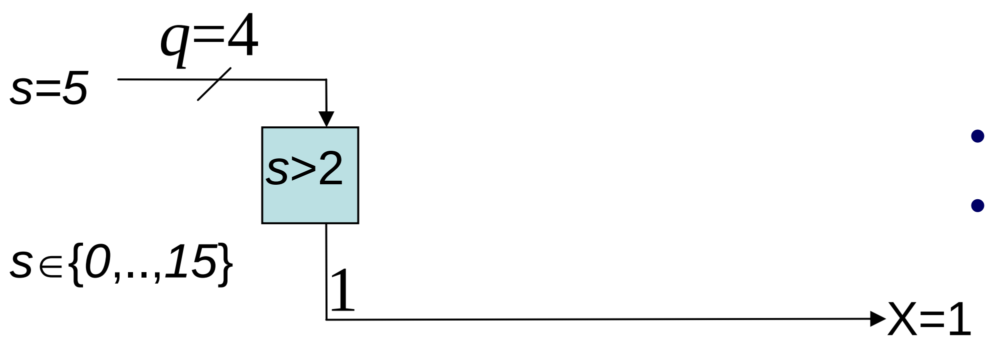
Non uniform random variable !

- Associate to each color a number (B=0, R=1, G=2, Y=3)
  => non uniform distribution.

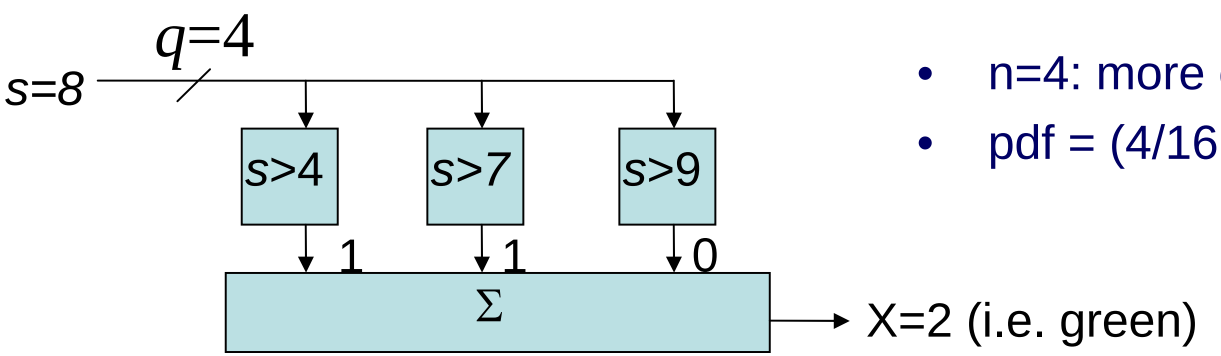
- By appropriate coloring of the face of the dice, any *n*=4 value distribution can be obtained (with a precision of 1/16).

Naive method to generate a discrete variable

- ELECTRONIC version: generation of a pseudo random number *s* between 0 and 15.
- The value of s is compared to *n*-1 threaholds to determine the output value.



- n=2: very simple hardware
- Here : P(X=0) = 2/16, P(X=1)=14/16



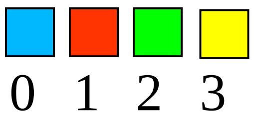
- n=4: more complex hardware
- pdf = (4/16, 3/16, 2/16, 7/16)

- PROBLEM : If *n*=8 => 255 comparateurs in parallel => High complexity

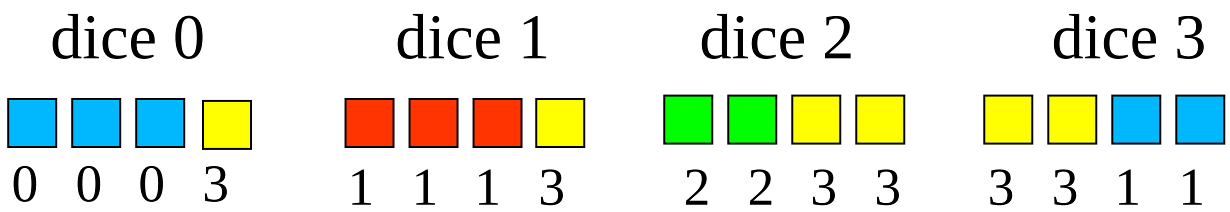
The Walker 's Alias method

- **Replace a single throw of a 16-face dice with 2 throws of 4-face dices...**

  - First throw: Select randomly a color among the 4 (4 differents faces).



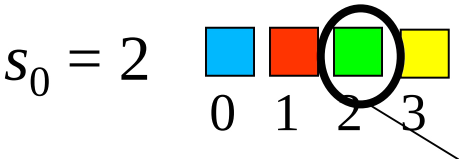
  - Second throw: according to the color, select the corresponding two color dice and throw it. Keep the last color (or number).



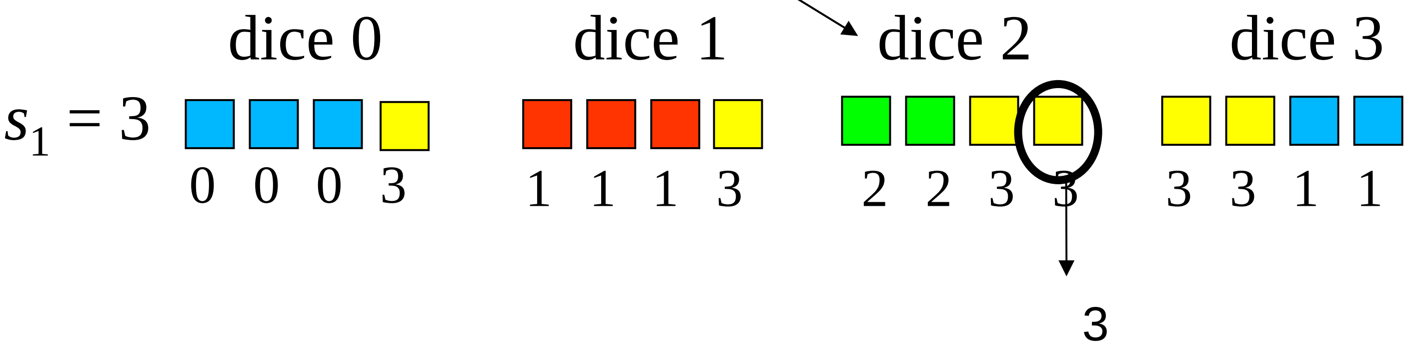
The Walker 's Alias method

- **Replace a single throw of a 16-face dice with 2 throws of 4-face dices...**

  - First throw: Select randomly a color among the 4 (4 differents faces).

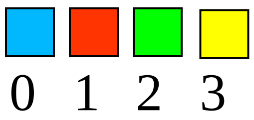


  - Second throw: according to the color, select the corresponding two color dice and throw it. Keep the last color (or number).

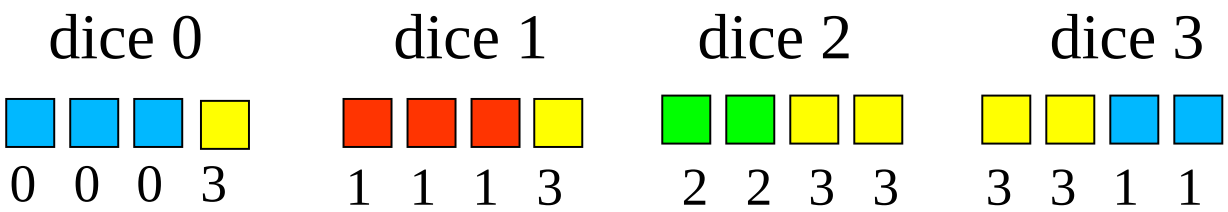


The Walker 's Alias method

- **Replace a single throw of a 16-face dice with 2 throws of 4-face dices...**
  - First throw: Select randomly a color among the 4 (4 differents faces).



  - Second throw: according to the color, select the corresponding two color dice and throw it. Keep the last color (or number).

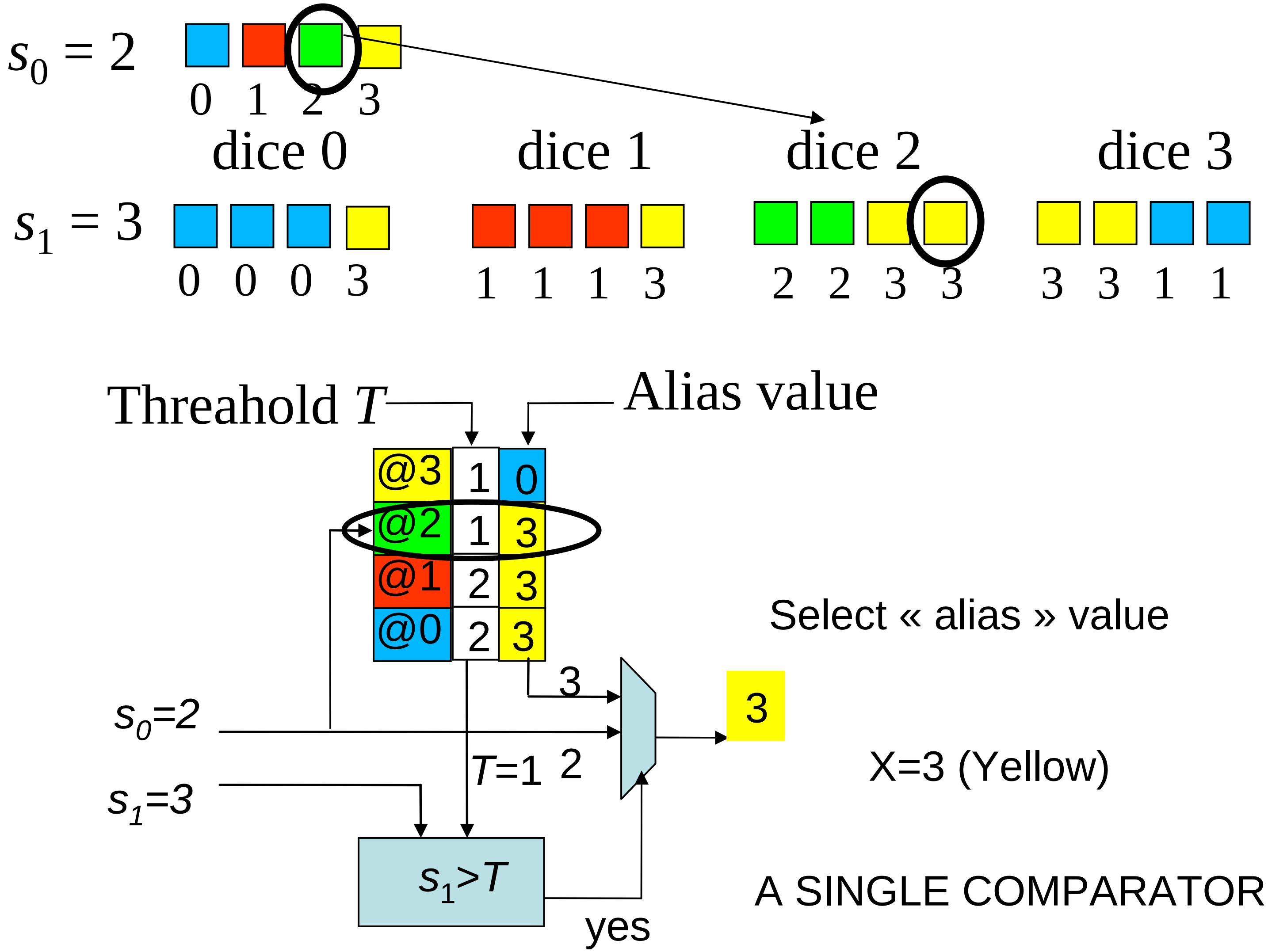


P(X=0) = 5/16 P(X=2) = 2/16

P(X=1) = 3/16 P(X=3) = 6/16

- We have the initial non-uniform distribution

The Walker 's Alias method

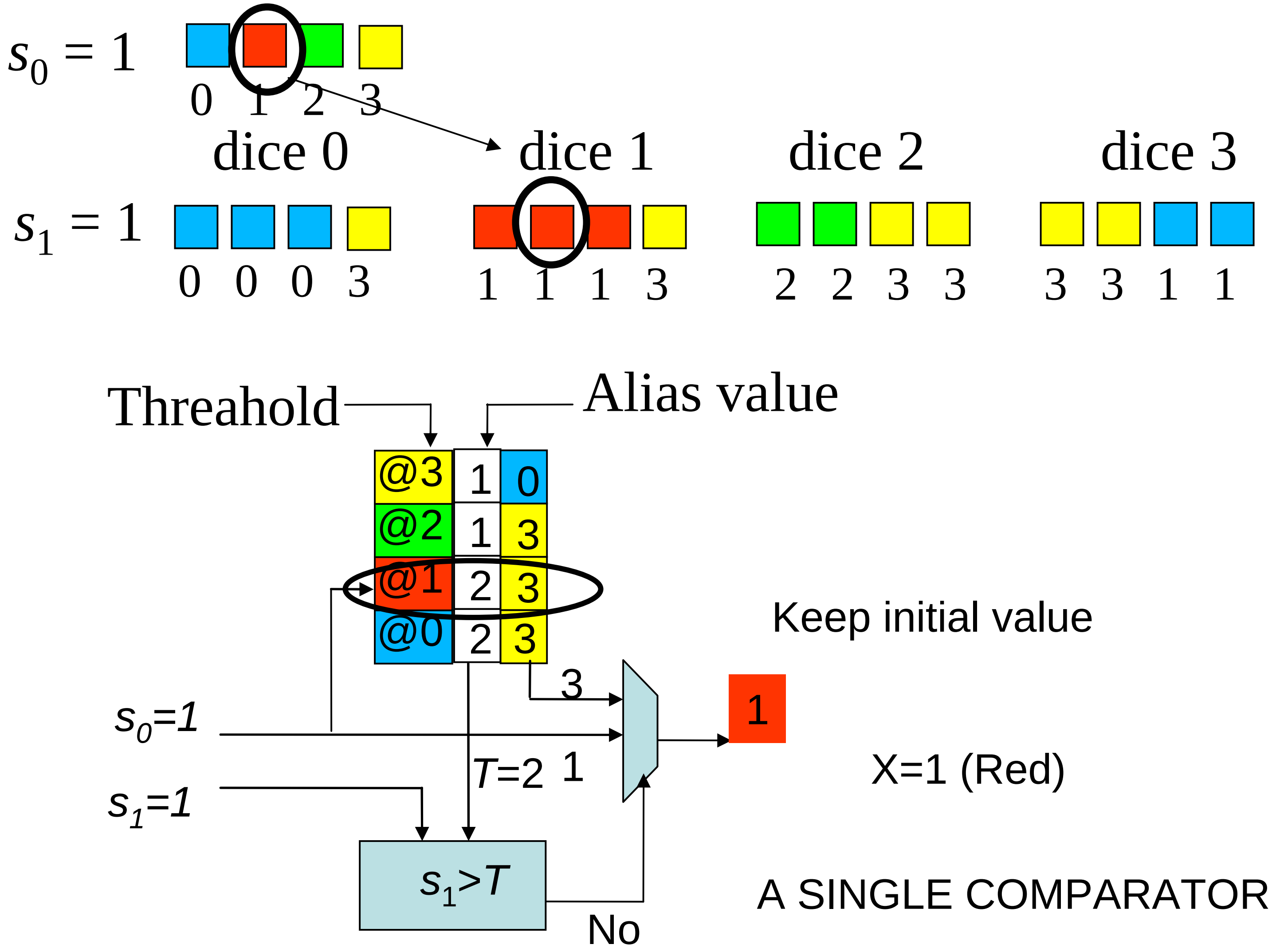


Select « alias » value

X=3 (Yellow)

A SINGLE COMPARATOR…

The Walker 's Alias method

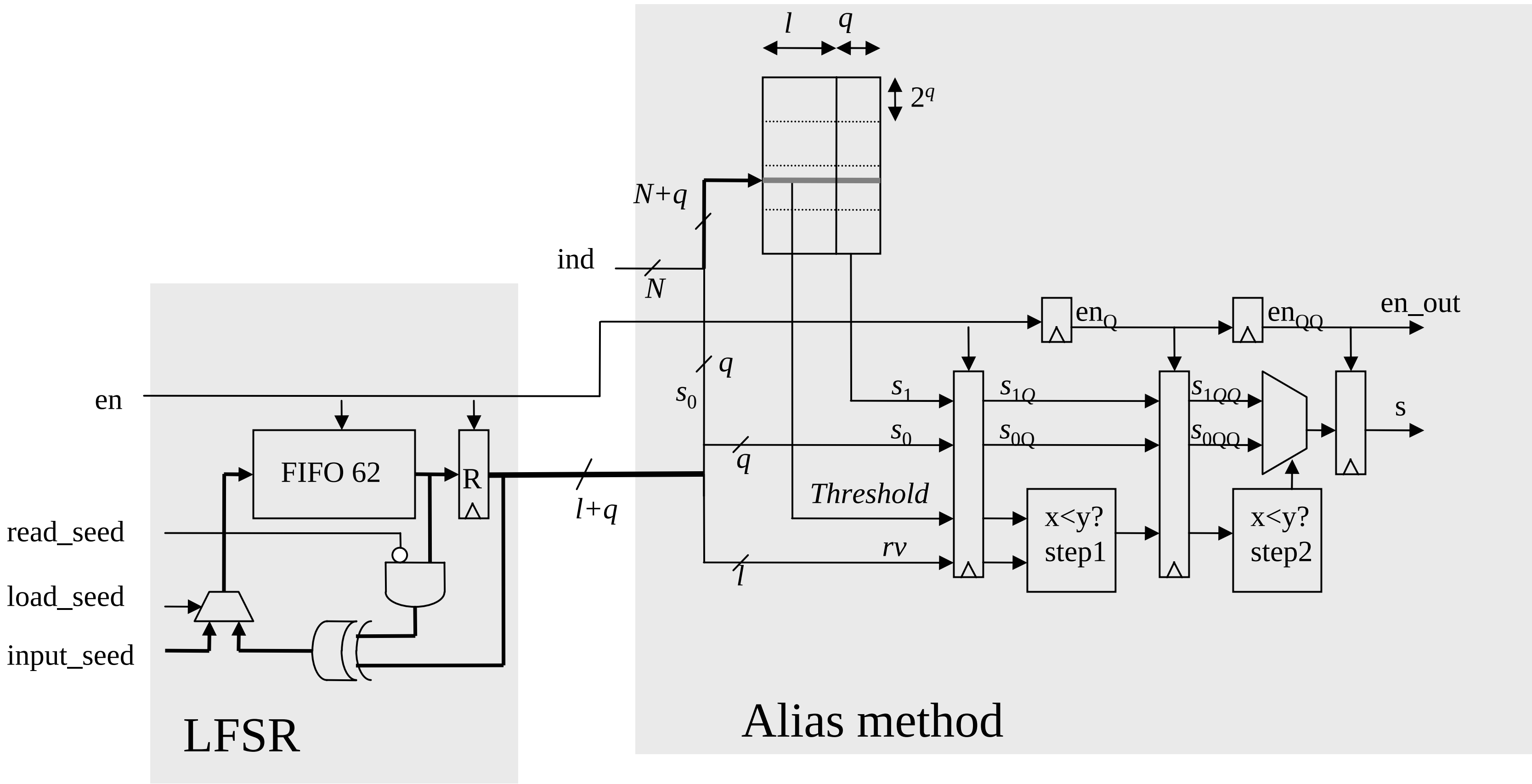


Keep initial value

X=1 (Red)

A SINGLE COMPARATOR…

Architecture of the HDCE



Functionalities

- **Generic VHDL**
- **From channel model => optimal generation (in sense of MSE) of alias table.**
  - Ad-Hoc solution
- **Seed management functionnality of random generator**
  - Compatible Read/Write process in Hardware and Solftware
- **C and VHDL fully compatible**
  - Both Observability and high speed simulation.
  - Compatible with the RMCS method

CONCLUSION

- **PROPOSITION OF TWO “PRAGMATIC” METHODS TO AVOID “BRUT FORCE” MONTE-CARLO SIMULATIONS**
  - Reduced Monte Carlo Simulation
  - Entropy Inspired Distance
- **THOSE TWO METHODS ALLOW TO TRADE-OFF “TIME OF SIMULATION” VERSUS “PRECISION”**
  - MONTE-CARLO SIMULATION is the “final judge”
  - => HDCE for hardware high speed MC Simulation
- **WE HAVE (WILL) COMBINED THOSES 3 METHODS TO OPTIMIZE THE DECODER ARCHITECTURE OF NON BINARY LDPC CODE OF THE DAVINCI PROJECT.**

PUBLICATIONS

- **REFERENCES :**
- **EID :**
- **A. Singh, A. Al-Ghouwayel1, G. Masera, E. Boutillon, " New Performance Evaluation Metric for Sub-Optimal Iterative Decoders", Accepted to IEEE Communications letters, nov. 2008.**

- **RMCS :**
- **E. Boutillon, C. Douillard, G. Montorsi, " Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues", Transactions of the IEEE, vol. 95, n°6, june 2007.**

- **http://www.ict-davinci-codes.eu/**