800MHz 12D 12PAM Mapping and Frame Structure Proposal

IEEE P803.2an Task Force

Ottawa, September ’04

Brett McClellan

Supporters

- George Zimmerman — Solarflare
- Katsutoshi Seki — NEC
- Jose Tellado — Teranetics
- Chine-Hsin Lee — KeyEye

Problem Statement

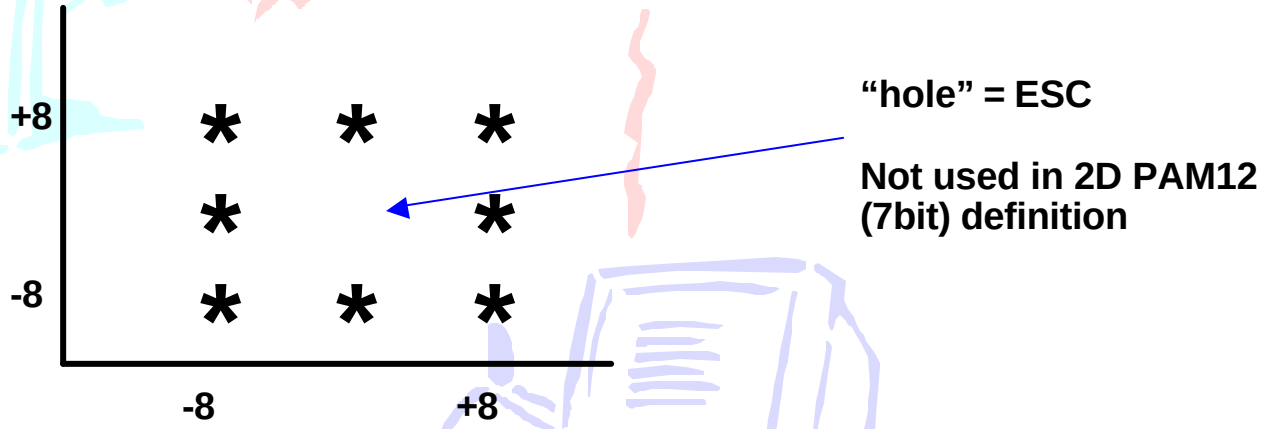
- **The proposed 2D 12PAM constellation is an inefficient mapping and results in both a power and baud rate penalty.**
  - **“donut” constellation has 0.43dB higher power than uniform 12PAM**
  - **>2% baud rate penalty vs. a square constellation**
- **The proposed frame structure is also inefficient and very complex**
  - **51,200 payload bits over 4224 baud**

12D 12PAM Mapping

- 12-dimensional mapping
  - 4 pairs x 3 baud = 12D constellation
- 43 bits per mapping (3 baud)
  - 14.3333 bits per baud vs. 14
  - 8 coded + 6 1/3 uncoded bits per baud
  - 24 coded + 19 uncoded bits per 12D symbol
- Uses the full 12x12 PAM constellation
  - viewed from any 2D
- Only 0.03dB TH precoding loss
- Transmit power independent of line length

12D Mapping Construction

- **Use the proposed "donut" as a base mapping for the 2D constellations**
  - **described in Powell (July)**
  - **Every 7 bits are mapped to 2 pair (2D)**
    - **4 coded bits to select coset**
    - **3 uncoded bits to select 1 of 8 partitions**



- **Treat the missing 9th partition as ESC**

12D Mapping Construction

- **Create 12D symbol from 6 sets of 2D mappings**
  - **AB1, CD1, AB2, CD2, AB3, CD3**
  - **43-bit input ® 1bit MSB + 6 x 7bits**
    - **for MSB=0 use the base mappings for all 6 2D groups**
    - **for MSB=1 replace some 2D mappings with ESC**
    - **19 uncoded bits select the 6 2D partitions (see table T1)**
      - **The 6 2D mappings may contain 1, 2 or 3 ESC symbols based on the word value**
    - **coded bit mappings remain unchanged**
      - **24-bits ® 2 bits per pair per baud**
  - **12D Mapping is described in 4 tables**

12D ESC Mapping Tables

- **Table T1**

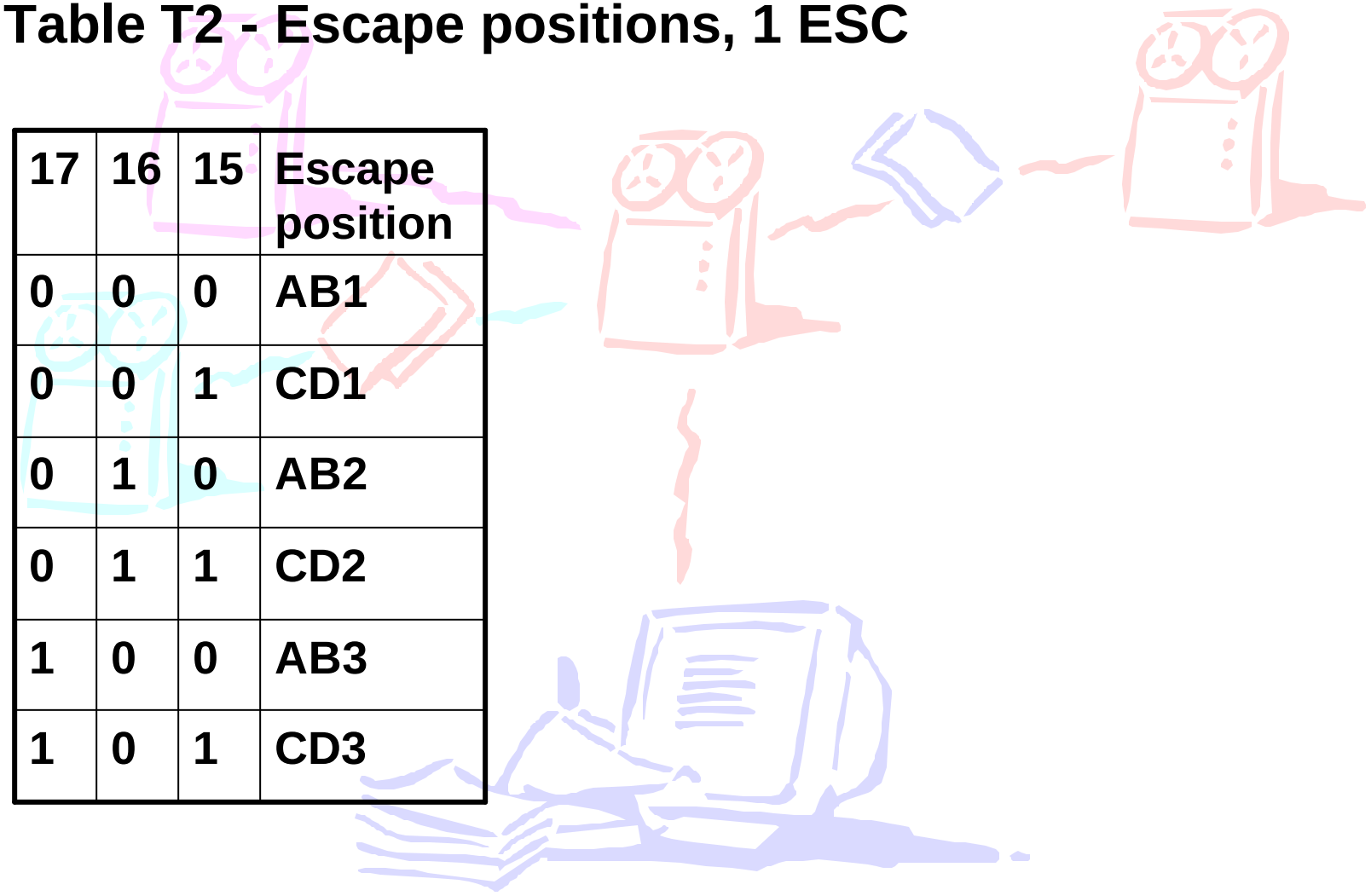
  19 uncoded bits select the 6 2D partitions

<table>
<tr><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td colspan="3">AB1</td><td colspan="3">CD1</td><td colspan="3">AB2</td><td colspan="3">CD2</td><td colspan="3">AB3</td><td colspan="3">CD3</td></tr>
<tr><td>1</td><td>0</td><td colspan="2">1 x ESC
See T2</td><td colspan="14">5 x 2DPAM12, in order as above in the remaining 2D partitions</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>see T2</td><td colspan="14">5 x 2DPAM12, in order as above in the remaining 2D partitions</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="3">2 x ESC
See T3</td><td colspan="12">4 x 2DPAM12, in order as above
in the remaining 2D partitions</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="2">3 x ESC
See T4</td><td colspan="10">3 x 2DPAM12, in order as above in the
remaining 2D partitions</td></tr>
</table>

12D ESC Mapping Tables

Table T2 - Escape positions, 1 ESC

17	16	15	Escape position
0	0	0	AB1
0	0	1	CD1
0	1	0	AB2
0	1	1	CD2
1	0	0	AB3
1	0	1	CD3



12D ESC Mapping Tables

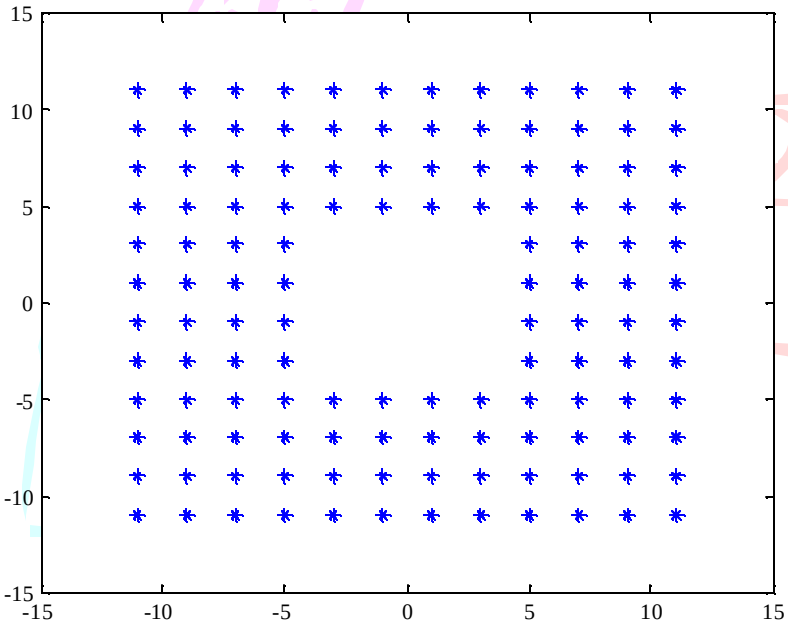
T3 - Escape positions, 2 ESC

15	14	13	12	Escape positions
0	0	0	0	AB1, CD1
0	0	0	1	AB1, AB2
0	0	1	0	AB1, CD2
0	0	1	1	AB1, AB3
0	1	0	0	AB1, CD3
0	1	0	1	CD1, AB2
0	1	1	0	CD1, CD2
0	1	1	1	CD1, AB3
1	0	0	0	CD1, CD3
1	0	0	1	AB2, CD2
1	0	1	0	AB2, AB3
1	0	1	1	AB2, CD3
1	1	0	0	CD2, AB3
1	1	0	1	CD2, CD3
1	1	1	0	AB3, CD3

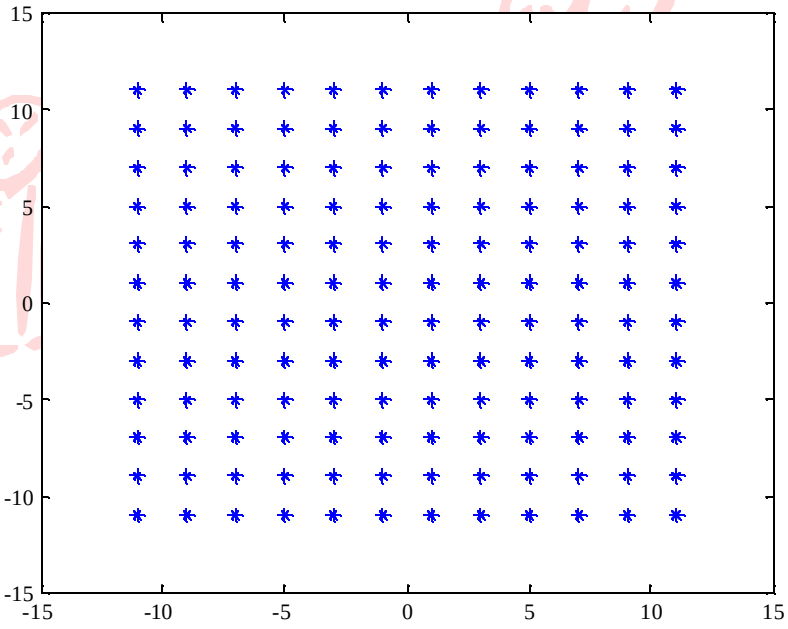
T4 - Escape positions, 3 ESC

11	10	9	Escape positions
0	0	0	AB1, CD1, AB2
0	0	1	CD2, AB3, CD3
0	1	0	AB1, CD1, AB3
0	1	1	AB2, CD2, CD3
1	0	0	AB1, AB2, CD2
1	0	1	CD1, AB3, CD3
1	1	0	AB1, AB2, CD3
1	1	1	CD1, CD2, AB3

2D Constellation View



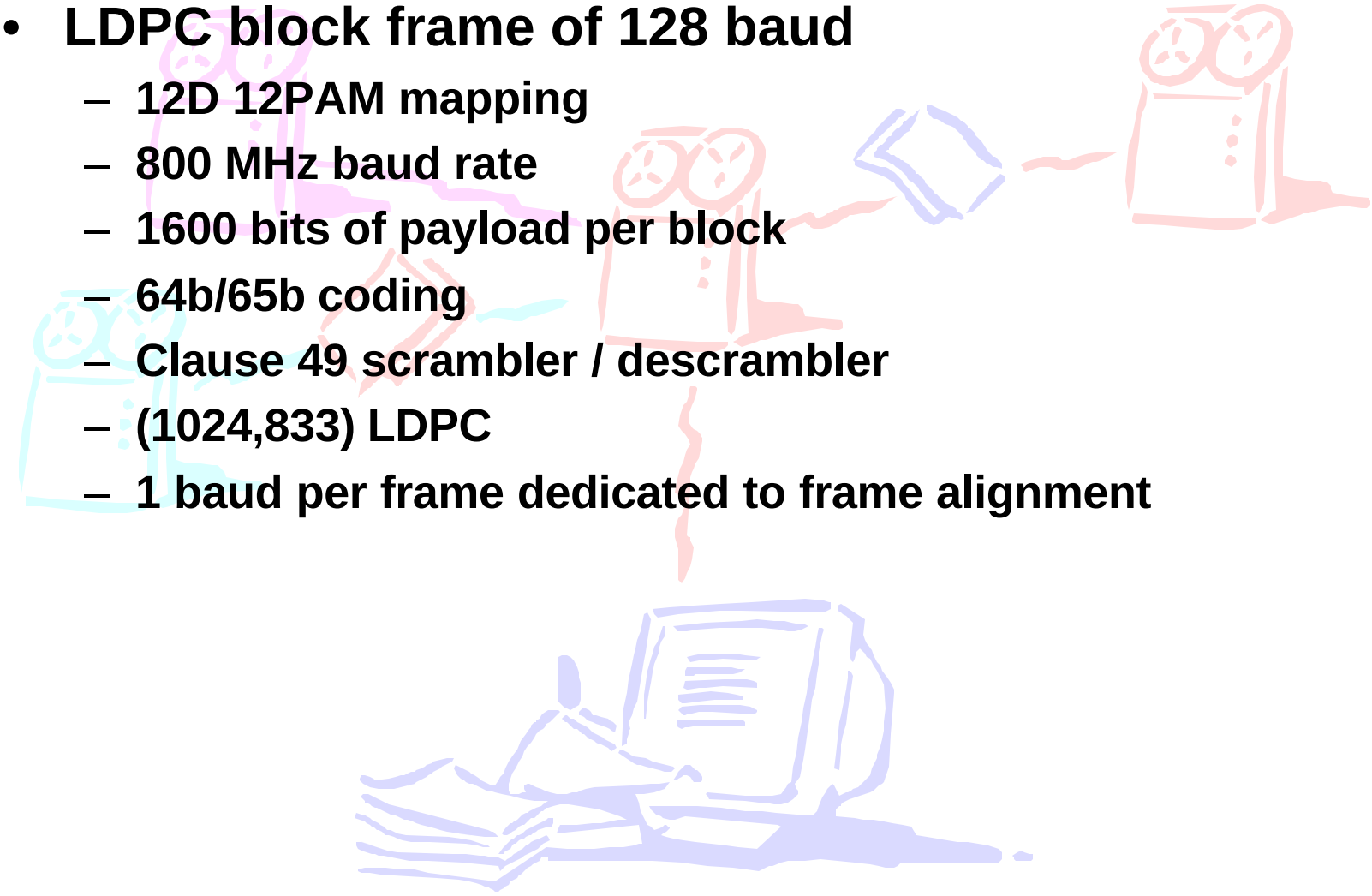
2D 12PAM “donut” constellation



2D view of the 12D 12PAM constellation

Frame Structure for 12D 12PAM

- **LDPC block frame of 128 baud**
  - **12D 12PAM mapping**
  - **800 MHz baud rate**
  - **1600 bits of payload per block**
  - **64b/65b coding**
  - **Clause 49 scrambler / descrambler**
  - **(1024,833) LDPC**
  - **1 baud per frame dedicated to frame alignment**



Frame Construction

- Construct an LDPC block from 25 blocks of 64b/65b coded data
  - 50 XGMII transfers -> 1600 bits
  - 1600 bits->65/64 -> 1625 bits
  - Add 4 pad bits -> 1629 bits
- Split into coded/uncoded
  - 1629 -> 825 coded + 804 uncoded
- LDPC encode using (1024,833)
  - Use 825 bits + 8 pad bits (= ‘0’)
  - Generate 191 check bits

Frame Construction

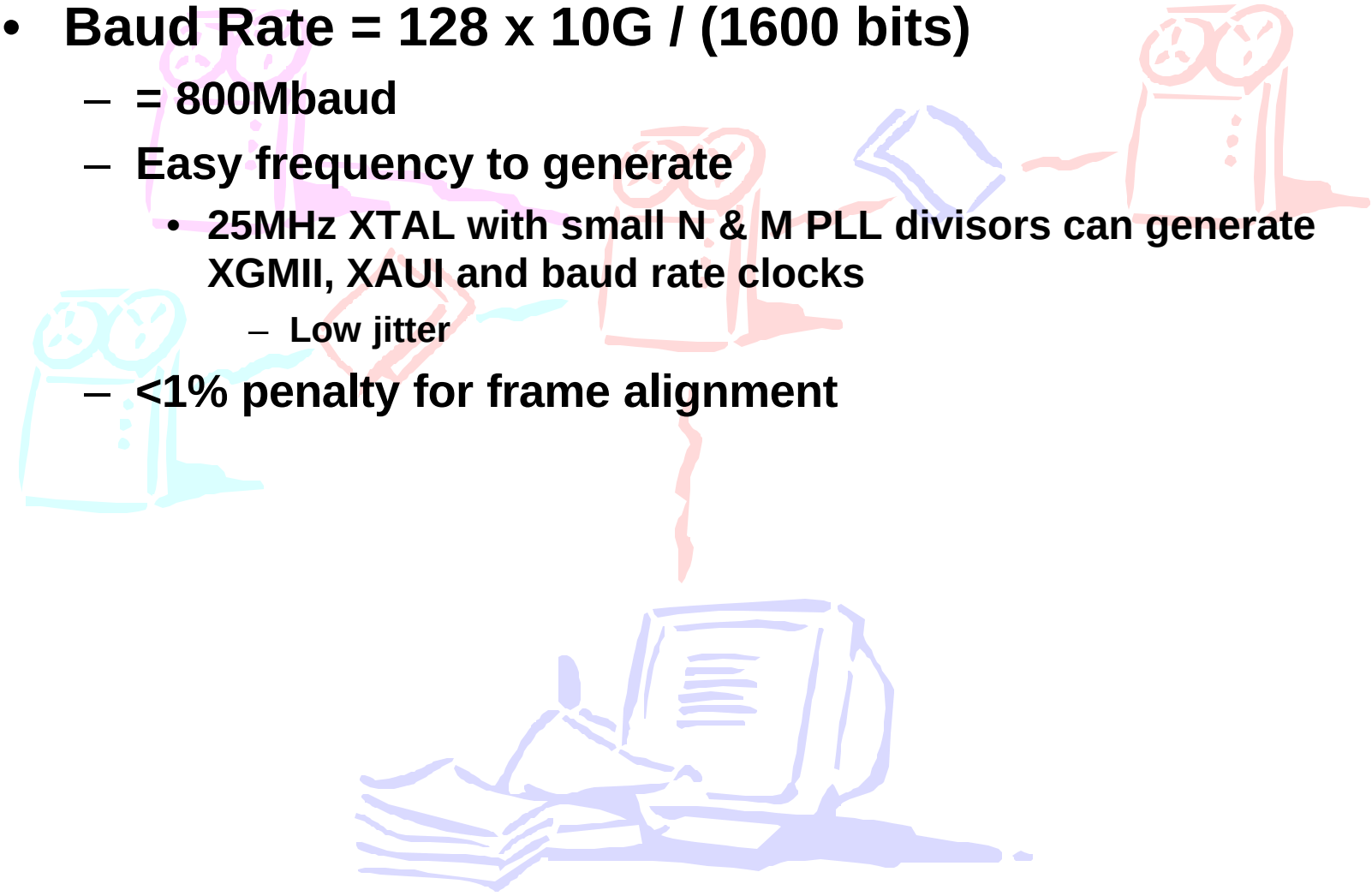
- **LDPC encoded block**
  - **804 uncoded bits**
  - **825 coded bits + 191 check bits = 1016 bits**
- **Transmit bits over 127 baud frame**
  - **127 baud of 12PAM carries 127 x 8 = 1016 coded bits**
  - **126 baud of 12D 12PAM + 1 baud of 4D 12PAM (2 pairs of 2D) carries 126x19/3 + 6 = 804 uncoded bits**

Frame Alignment

- **A single baud is added to every 127 baud LDPC block for frame alignment**
  - **2PAM: e.g. +7/-7**
  - **An alternating or PN sequence may be transmitted during this time for alignment testing**
  - **Or, the LDPC blocks may be grouped into bundles of 8 preceded by an 8 x 4D 2PAM frame alignment symbol (per the Tellado_1_0704 12PAM proposal)**

Baud Rate

- **Baud Rate = 128 x 10G / (1600 bits)**
  - **= 800Mbaud**
  - **Easy frequency to generate**
    - **25MHz XTAL with small N & M PLL divisors can generate XGMII, XAUI and baud rate clocks**
      - **Low jitter**
  - **<1% penalty for frame alignment**



Lower baud rate option

- Frames are 127 baud
- Perform frame alignment during the startup process per the 8PAM proposal.
- Alignment status is determined by LDPC decoder
- Baud Rate = 127 x 10G / (1600 bits)
  - = 793.75 Mbaud
    - Uses 25 MHZ XTAL -> (25 * 127 / 4)
  - ~ 1% lower than 800MHz

Additional Codes

- **Extensible to other codes and block sizes**
  - **e.g. (2048,1723) with 255 baud block @ 781.25 MHz**

12D 12PAM baud rate calculations

LDPC out	**1024**	**2048**
LDPC in	**833**	**1723**
# baud	**127**	**255**
pad bits	4	15
payload (bits)	1600	3264
baud rate (MHz)	**793.75**	**781.25**
SNR	30.07	30.37
SNR gain vs 825MHz	**0.79**	**1.09**
add 1 baud per frame		
baud rate (MHz)	800	784.313725
SNR	29.92	30.29
SNR gain vs 825MHz	**0.64**	**1.01**

Summary

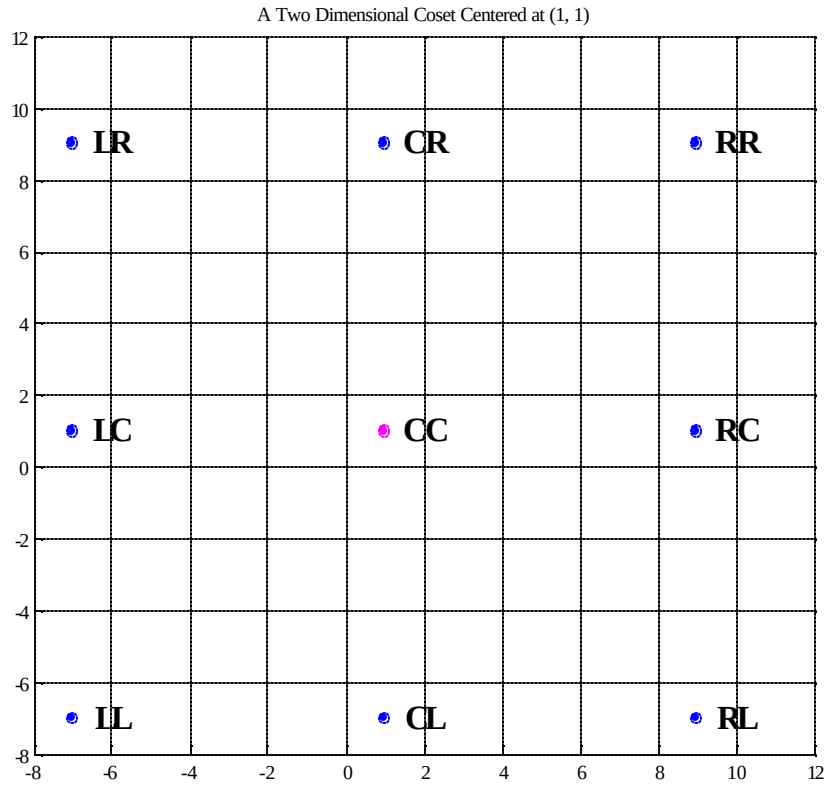
- **This 12D 12PAM mapping and framing proposal provides for:**
  - **800MHz baud rate using (1024,833) LDPC**
  - **0.64 dB SNR gain over prior proposal**
  - **Simple frame construction**
  - **128 baud frames**
  - **1600 bit payload**
  - **Robust frame alignment**
  - **Addresses the baud rate and transmit power penalties of the current 12PAM proposal.**
  - **Utilizes Clause 49 coding and scrambling**

Backup Slides

- **Alternative 12D Mappings**
- **Base-3 Mapping (Jose Tellado)**
  - **For each 12D symbol convert the 19 uncoded bits to 12 base-3 values**
  - **Each base-3 value selects a ternary (L,C,R) partition on a single pair per baud**
  - **Demapping uses reverse process**
- **Refinement to the 12D ESC (Dariush Dabiri)**
  - **Equivalent to 12D ESC but requires only 1 table**
  - **Description follows (from Dariush)**

2D Partitioning

- **Decompose the 2D constellation into two sets:**
  - **B: The 8 blue points.**
  - **R: The red point at the center.**



4D Labeling Rules

- Any 4D coset, S, has 3^4 = 81 points.
- Any 4D coset can be decomposed as:
  - S = X + Y + Z (see the table below).
- Points of X use 2 copies of the 3-bit Gray labeling of B. Each copy is used for 2D. A total of 6bits/4D.
- For points of Y, the MSB chooses between RxB and BxR. Use the 3-bit Gray labeling of B to choose points inside RxB (or BxR) (Note: R has only one point). A total of 4bits/4D.

Set	Formula	Points	No of Bits
X	B x B	64	6
Y	RxB + BxR	16	4
Z	R x R	1	0

12D Labeling

- **Each constellation subset is identified by a unique prefix (see the next table).**
- **Points inside each subset are labeled by 3 copies of the 4D labeling of the previous slide. The generated bits are called the ‘suffix bits’ of each constellation point.**
- **The next table shows prefixes for the 12D subsets used in our labeling:**

Prefix	Subset	No of Suffix Bits	Total Bits
0*	(X, X, X)	18	19
100*	(X, X, Y)	16	19
101*	(X, Y, X)	16	19
110*	(Y, X, X)	16	19
11100*	(Y, Y, X)	14	19
11101*	(Y, X, Y)	14	19
11110*	(X, Y, Y)	14	19
1111100*	(X, X, Z)	12	19
1111101*	(X, Z, X)	12	19
1111110*	(Z, X, X)	12	19
1111111*	(Y, Y, Y)	12	19

Mapper and Demapper

- **Mapper:**
  - **Prefix match to find the unique matching prefix and the corresponding subset.**
  - **Use the suffix bits to specify the constellation points according to the 4D labeling rules.**
- **Demapper:**
  - **Identify the unique 12D subset for the received 12D symbol.**
  - **Use the table to find the unique prefix.**
  - **Use the 4D labeling rules to find the suffix bits.**