64b/66b coding update

Rick Walker, Birdy Amrutur, Tom Knotts

Agilent Laboratories, Palo Alto, CA

rick_walker@agilent.com

Richard Dugan

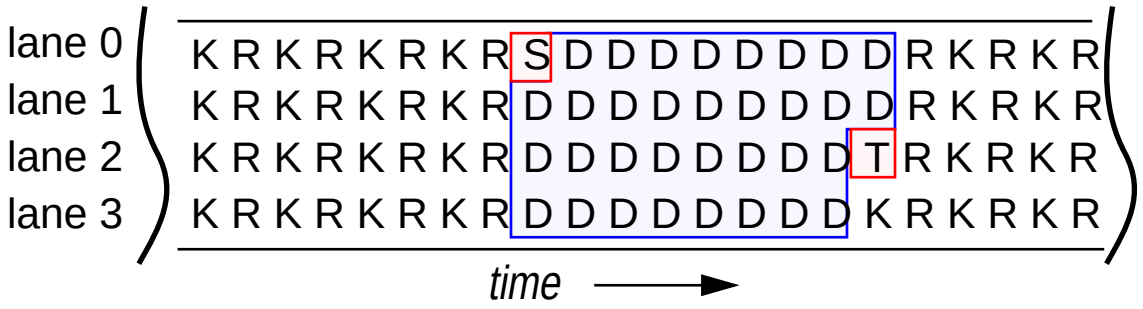
Agilent Technologies, Integrated Circuits Business Division, San Jose, CA

richard_dugan@agilent.com

Topics

- Code update
- Mean Time to False Packet Acceptance
- Coder Block Diagram and Gate Count
- Scrambler design
- Summary

Building frames with XAUI (HARI) mapping



S,T = SOP, EOP
K,R = control words (Z)
D = Data octets

pure data

D D
D D
D D
D D

D

pure control

Z Z
Z Z
Z Z
Z Z

Z

two possible packet startings

S D
D D
D D
D D

Z S
Z D
Z D
Z D

S

eight possible packet endings

T Z
Z Z
Z Z
Z Z

D Z
T Z
Z Z
Z Z

D Z
D Z
T Z
Z Z

D Z
D Z
D Z
T Z

D T
D Z
D Z
D Z

D D
D T
D Z
D Z

D D
D D
D T
D Z

D D
D D
D D
D T

T

Code Overview

Data Codewords have “01” sync preamble

Mixed Data/Control frames are identified with a “10” sync preamble. Both the coded 56-bit payload and TYPE field are scrambled

00,11 preambles are considered code errors and cause the packet to be invalidated by forcing an error (E) symbol on the HARI output

Code Summary

Hari Pattern	Sync		Bit fields 0-63							
DDDD/DDDD	0	1	D0	D1	D2	D3	D4	D5	D6	D7
ZZZZ/ZZZZ	1	0	0x1e	Z0	Z1	Z2	Z3	Z4	Z5	Z6	Z6
ZZZZ/SDDD	1	0	0x33	Z0	Z1	Z2	Z3		D5	D6	D7
SDDD/DDDD	1	0	0x78	D1	D2	D3	D4	D5	D6	D7
TZZZ/ZZZZ	1	0	0x87		Z1	Z2	Z3	Z4	Z5	Z6	Z7
DTZZ/ZZZZ	1	0	0x99	D0		Z2	Z3	Z4	Z5	Z6	Z7
DDTZ/ZZZZ	1	0	0xaa	D0	D1		Z3	Z4	Z5	Z6	Z7
DDDT/ZZZZ	1	0	0xb4	D0	D1	D2		Z4	Z5	Z6	Z7
DDDD/TZZZ	1	0	0xcc	D0	D1	D2	D3		Z5	Z6	Z7
DDDD/DTZZ	1	0	0xd2	D0	D1	D2	D3	D4		Z6	Z7
DDDD/DDTZ	1	0	0xe1	D0	D1	D2	D3	D4	D5		Z7
DDDD/DDDT	1	0	0xff	D0	D1	D2	D3	D4	D5	D6

There are three choices per bit, so frames can be composed with 64, 4:1 multiplexors controlled according to 1 of 12 frame types

Control code mapping

8B/10B	name	shorthand	7-bit line code
K28.0	idle1	R	0x00
K28.1	busy idle0	Kb	0x1e
K28.2	reserved0	-	0x2d
K23.7	busy idle1	Rb	0x33
K27.7	start	S	encoded by TYPE byte
K29.7	terminate	T	encoded by TYPE byte
K28.4	reserved1	-	0x4b
K28.5	idle0	K	0x55
K30.7	error	E	0x66
K28.7	reserved2	-	0x78

- The 7-bit line codes representing 8B/10B control characters have 4-bit minimum hamming distance.

False Packet Acceptance Rate

- A key parameter of any code is the rate at which “damaged” packets are accepted as valid. In general, such a failure is capable of hard crashing a computer system.
- For 1Gb Ethernet the Mean Time to False Packet Acceptance (MTTFPA) was calculated to be approximately 60 billion years.
- Because 64b/66b has a uniform 4-bit Hamming protection, a conservative estimate can be made. Assume that packets with four or more errors will generate a false packet acceptance event. In practice, this overestimates the failure rate by about 2^{32} .

False Packet Acceptance Rate

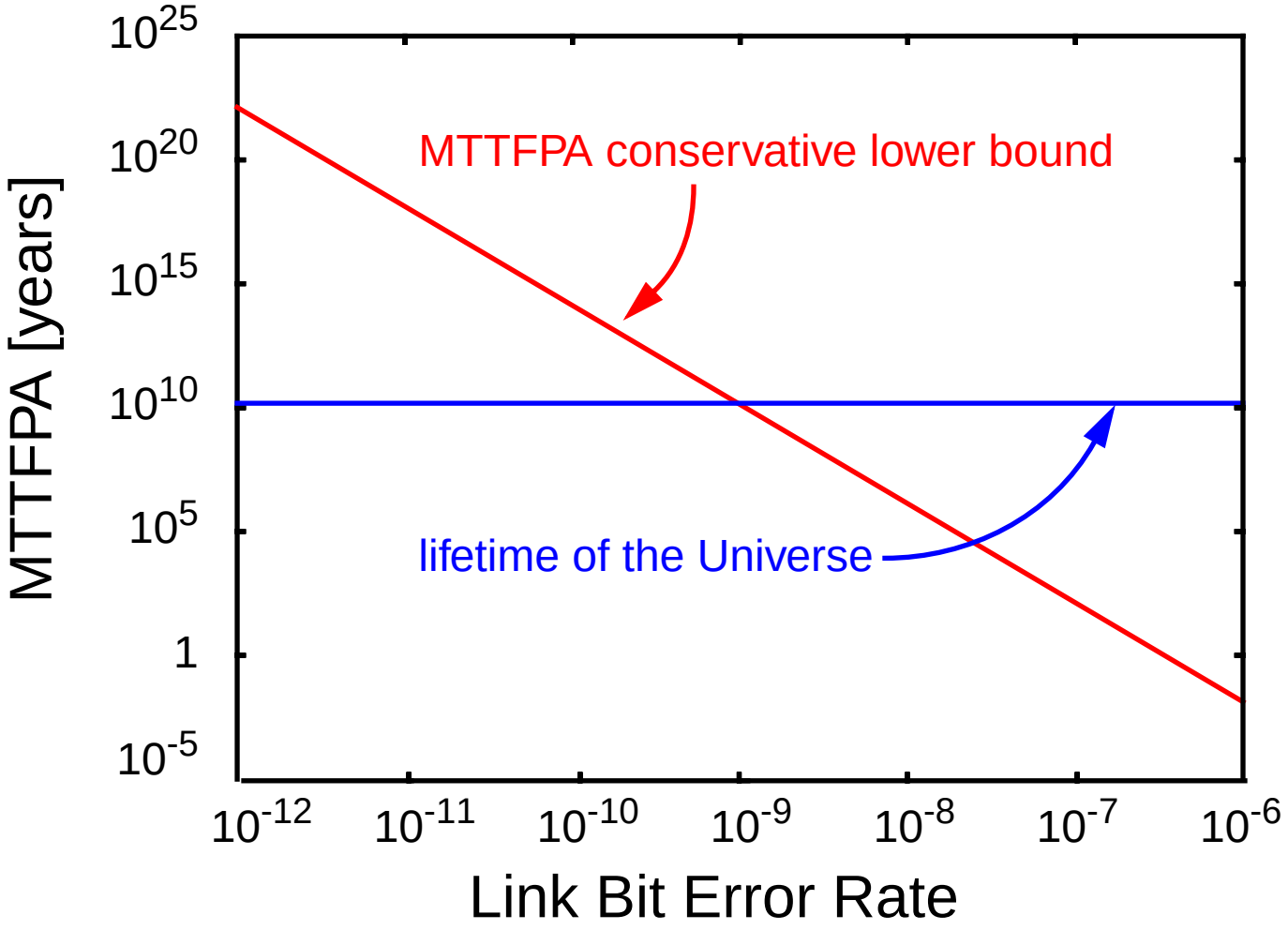
- P = coded packet size = 58+1526*8*66/64
- p_e = bit error rate, N = number of errors, t_b = bit time (1/10.3125G).
- probability of N errors in packet of size P:

$$p(N, P, p_e) = (1 - p_e)^{P-N} (p_e)^N \binom{P}{N}$$

- expected time for 4 or more errors:

$$MTTFPA > \frac{t_{bit} P}{1 - p(N, P, 0) - p(N, P, 1) - p(N, P, 2) - p(N, P, 3)}$$

False Packet Acceptance Rate

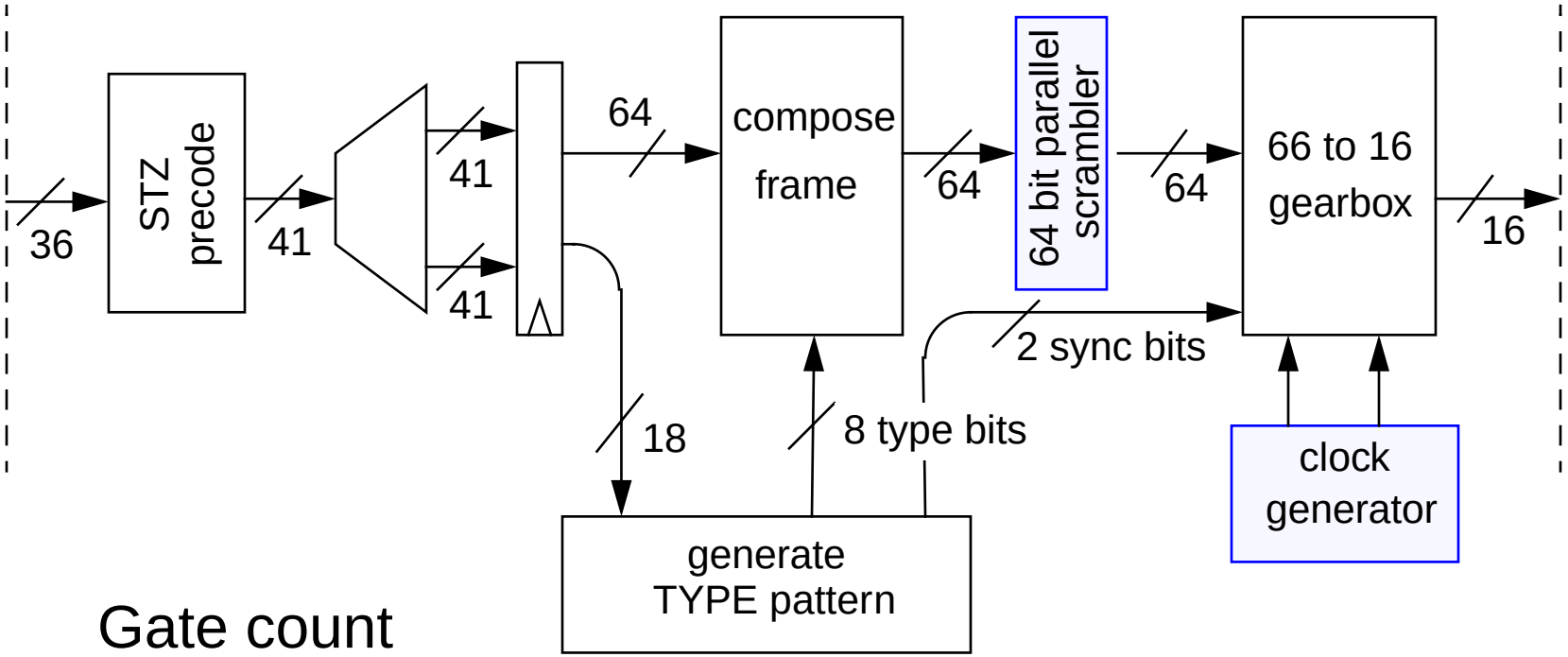


False Packet Acceptance Rate

Summary

- At a 10e-9 BER and 10.3Gb/s, the MTTFPA of 64b/66b is approximately equal to the 1Gb Ethernet 8b/10b performance at 10e-11 BER
- If 10G Ethernet maintains the same 10e-11 specification for PMD raw error rate, then 64/66b gives 7 orders of magnitude improvement in MTTFPA compared to coding used for 1G Ethernet

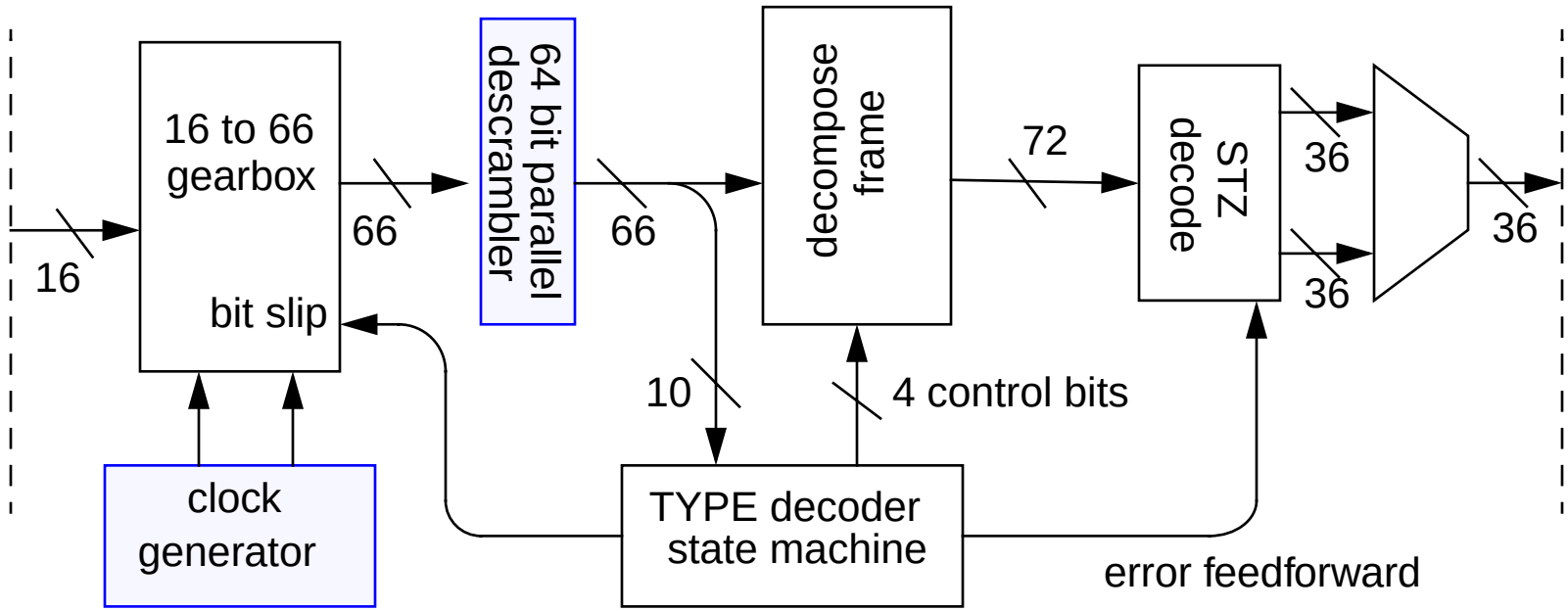
Coder Block Diagram



Gate count

- Logic: 731 logic cells + 234 flops
- Gearbox + clock generator: ~1400 flops

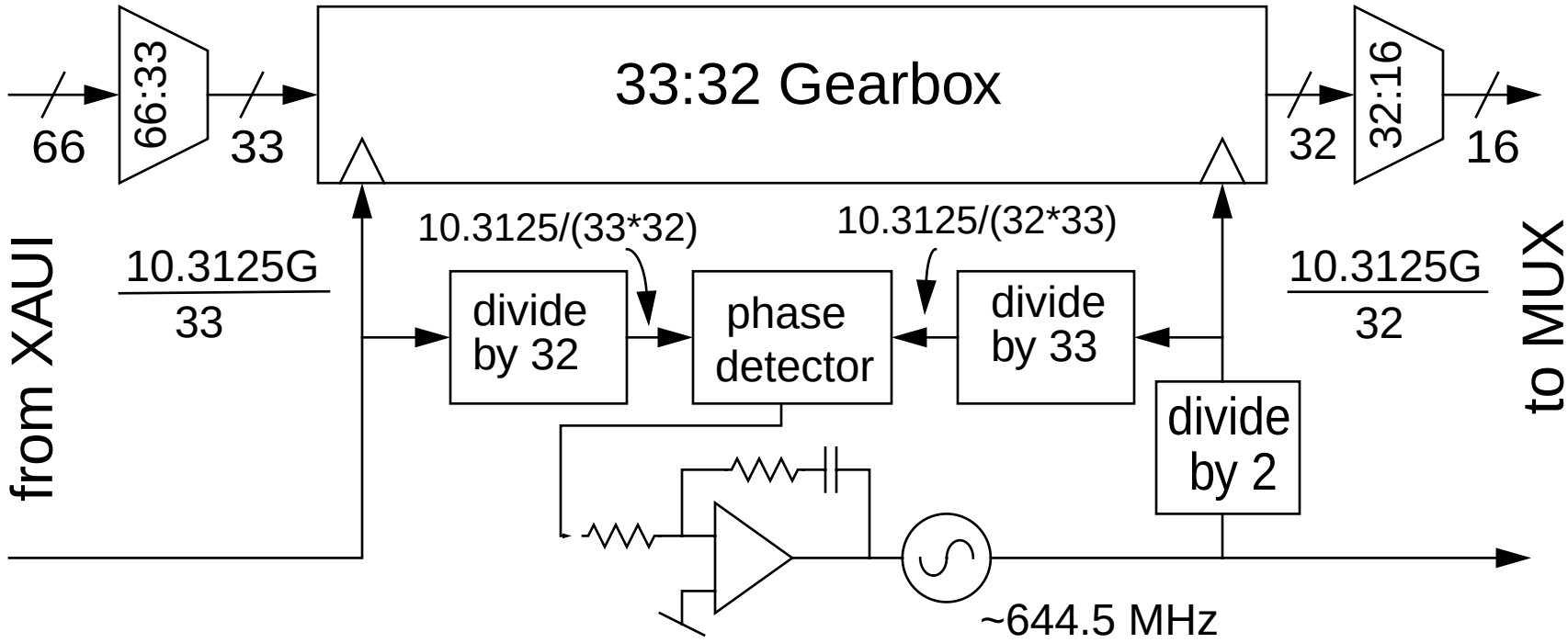
Decoder Block Diagram



Gate count

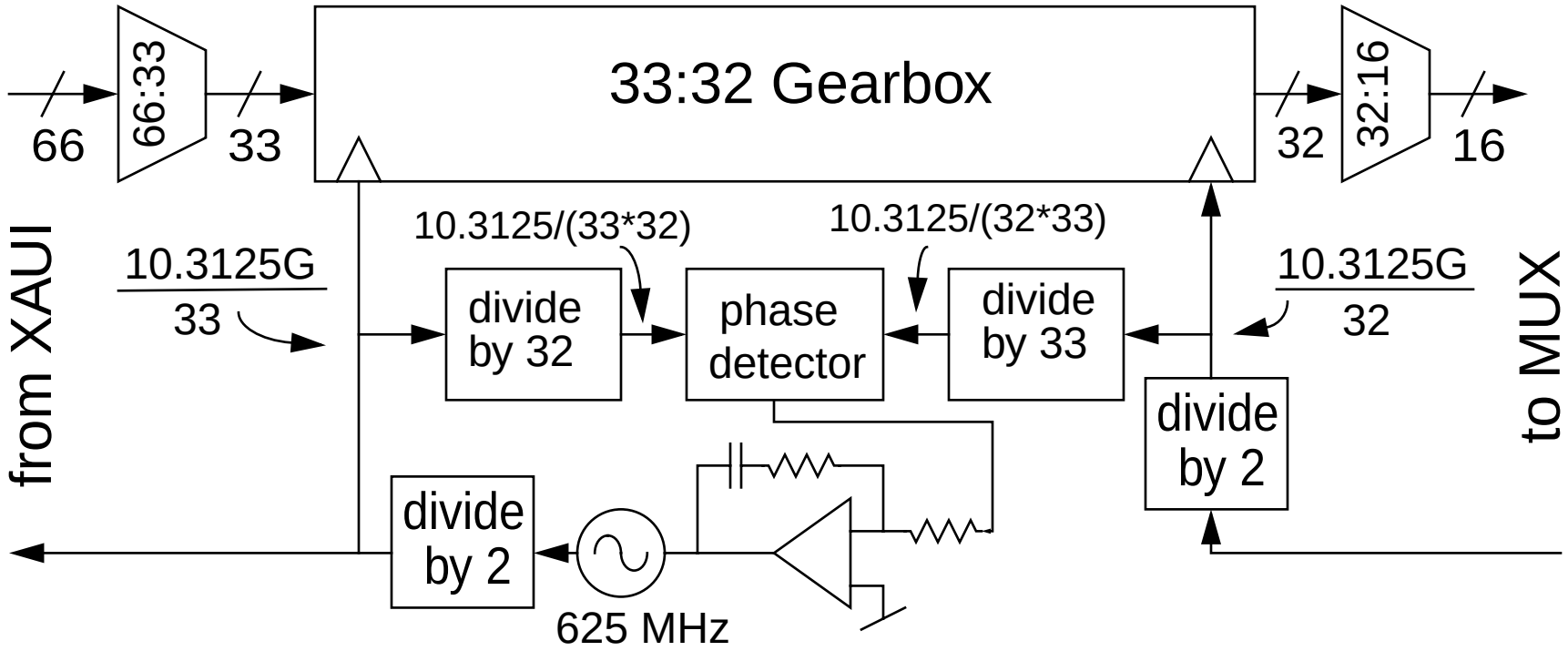
- Logic: 706 logic cells + 144 flops
- Gearbox + clock generator: ~1400 flops

TX Clock synthesis I



- PLL locks to received XAUI clock
- PLL provides clock to 16:1 MUX

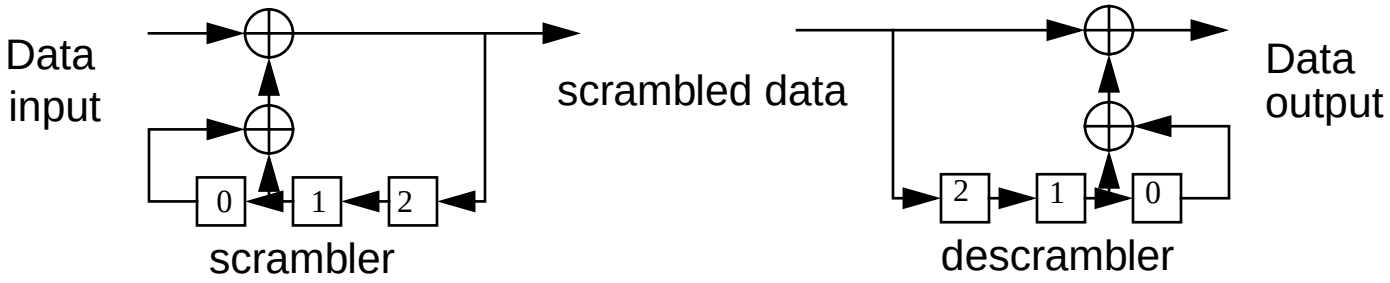
TX Clock synthesis II



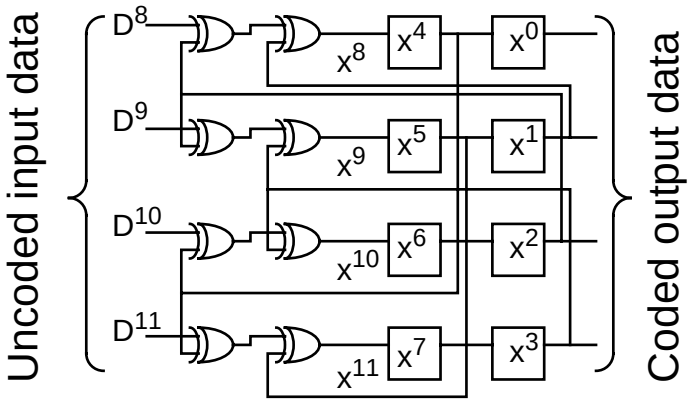
- PLL locks to MUX clock output
- PLL provides clock to XAUI output FIFO

Scrambling principle

An example 3-bit scrambler/descrambler in serial form:



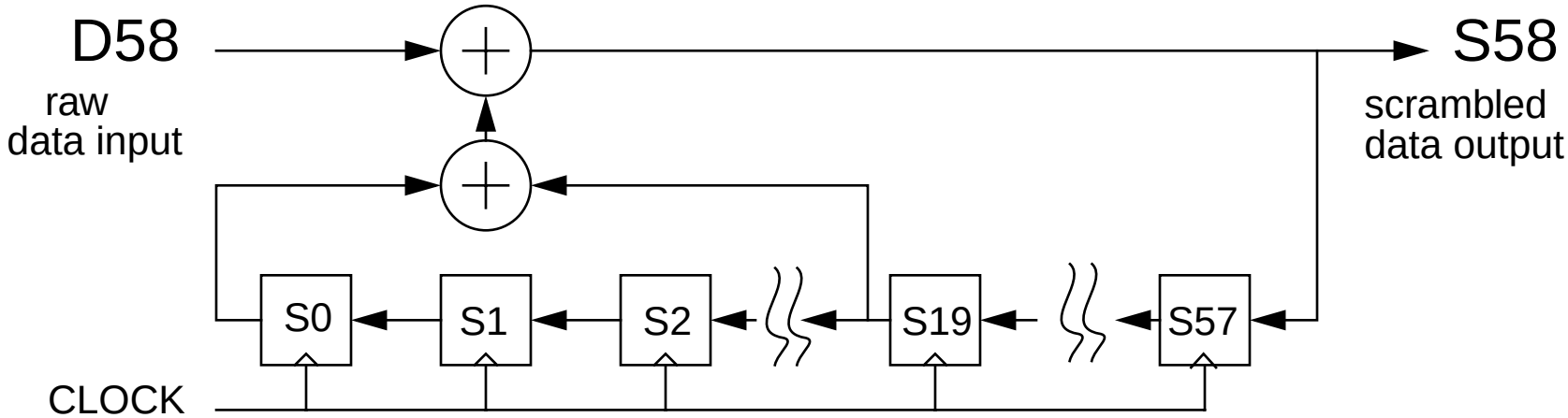
parallel form:



- Self synchronizing scrambler
- Can be parallelized for efficient implementation
- Using along pattern length reduces possibility of jamming (eg: $x^{58}+x^{19}+1=0$)
- Long pattern length self-synchronizing scramblers exist that do not compromise Ethernet CRC coverage

Derivation of Parallel Scrambler

Start with the Serial form of the Scrambler:



Write the recursion equation:

$S58 = D58 + S19 + S0$ or $D58 = S58 + S19 + S0$

notice that if we set the data input == 0, then we get
0 = S58 + S19 + S0, which is often written as: $X^{58} + X^{19} + 1 = 0$

Derivation of Parallel Scrambler

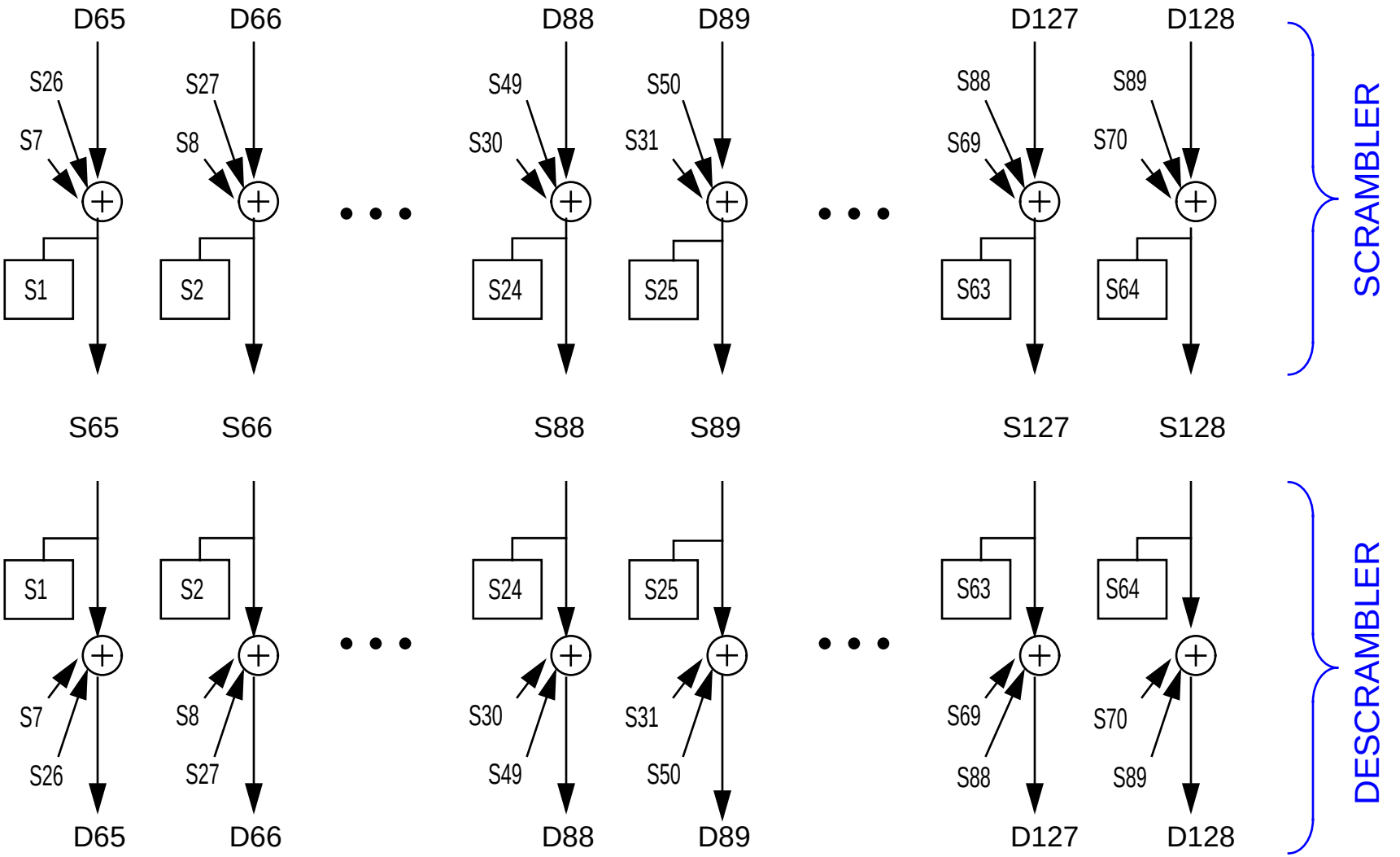
Use recursion equation to write terms for all parallel bits:

S58 = D58 + S19 +S0, so

S128 = D128 + S89 + S70
S127 = D127 + S88 + S69
S126 = D126 + S87 + S68
...
...
S66 = D66 + S27 + S8
S65 = D65 + S26 + S7

These equations are easily implemented with a parallel register and a handful of XOR gates.

Latency is equal to 2-cascaded, 3-input XOR delays, and 64 scrambled bits are computed all at once.


D65
D66
D88
D89
D127
D128
S26
S27
S49
S50
S88
S89
S7
S8
S30
S31
S69
S70
S1
S2
S24
S25
S63
S64
SCRAMBLER
S65
S66
S88
S89
S127
S128
S1
S2
S24
S25
S63
S64
S7
S8
S30
S31
S69
S70
S26
S27
S49
S50
S88
S89
D65
D66
D88
D89
D127
D128
DESCRAMBLER

Summary

- Since the last meeting, we've finished a gate level implementation and folded the results back into simplifying the code definition
- An analysis of Mean Time to False Packet Acceptance (MTTFPA) shows acceptable performance with 10e-9 BERs.
- Gate counts for 64/66 are reasonable and on par with the complexity of the four 8b/10b decoders need for XAUI
- Clock Generation is straightforward and easy in any process capable of XAUI PLL performance