Error Propagation with DFE's.

Mike Dudek QLogic

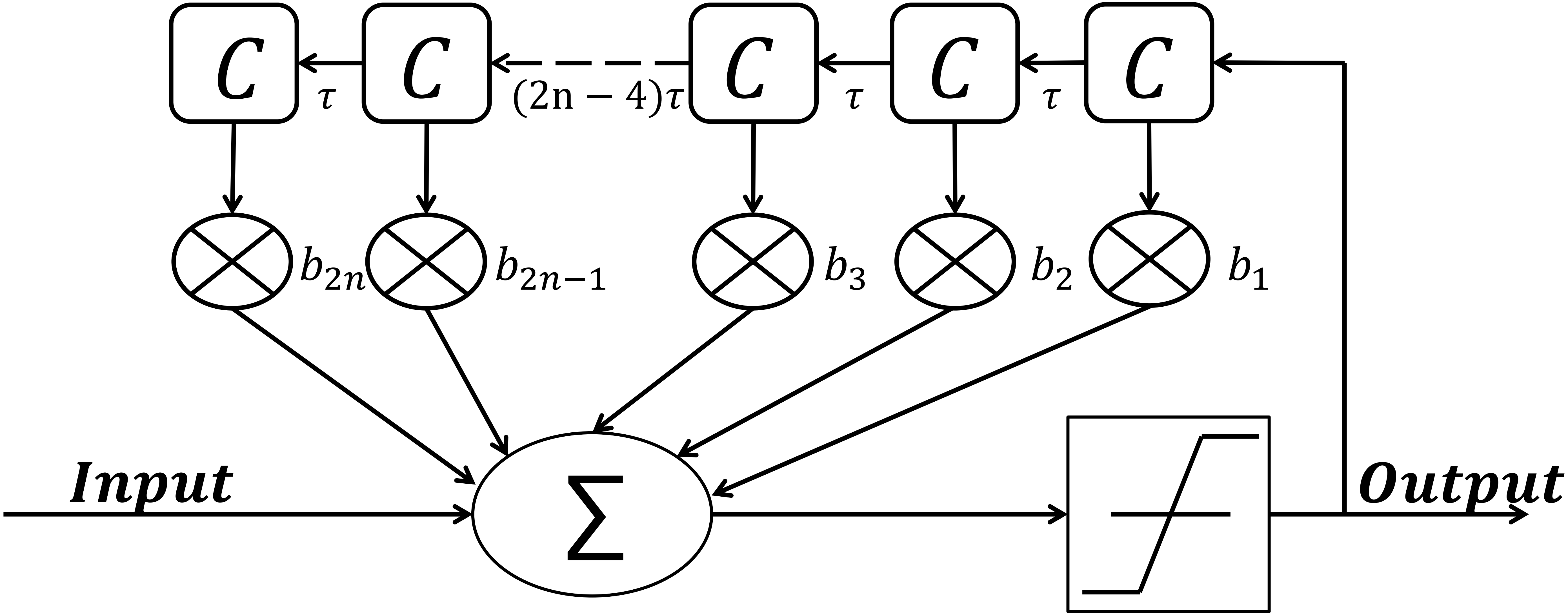
IEEE 802.3by

Introduction

QLOGIC

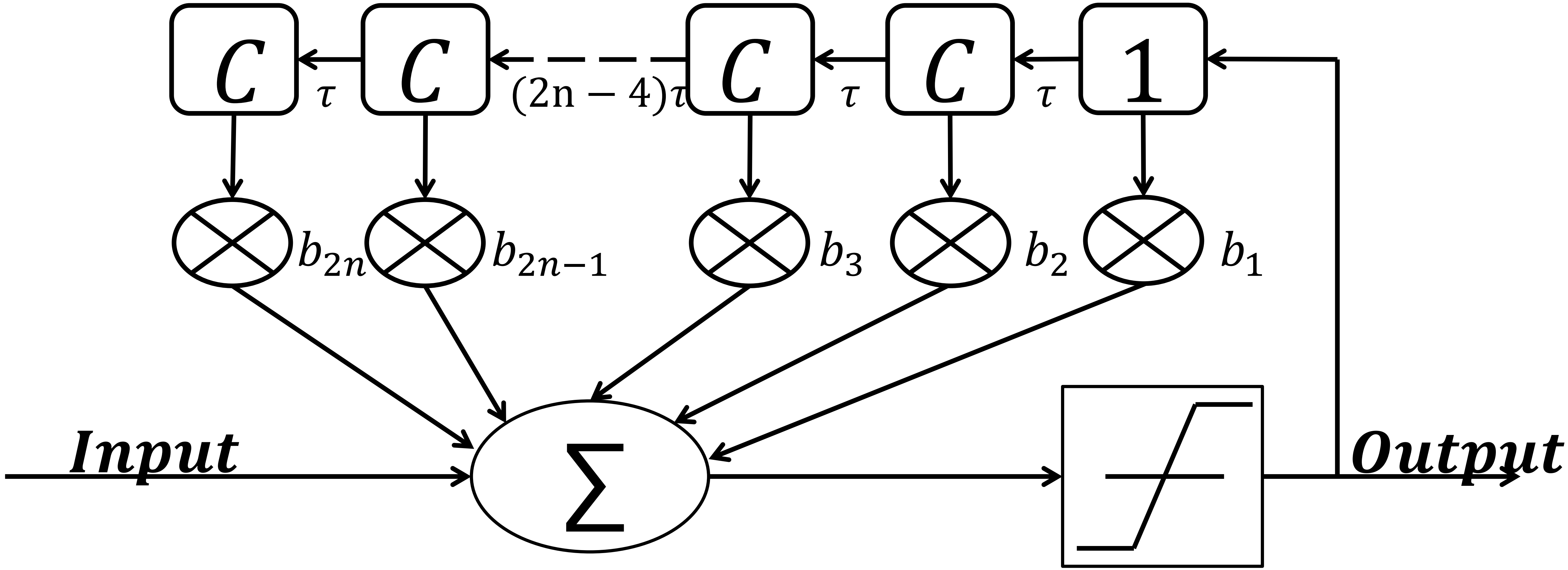
- Error Propagation with the use of DFE’s is a known issue that can create error bursts.   This presentation explores the error propagation mechanism and suggests what criterion could be used to reduce this effect.
- A version of this presentation was originally presented to the 25G Architecture ad-hoc. However at that ad-hoc it was pointed out that for the no-FEC case, problems can occur with much shorter error bursts making the original proposed change not appropriate for no-FEC operation. The presentation is amended based on this and no proposal for a change is being made.
- For ease of explanation it is assumed in the following that the initial error is a “zero” erroring to a “one”.   The analysis is the same for a “one” erroring to a “zero” but trying to consider both at the same time makes explanation more difficult.
- For the initial analysis it is assumed that the first tap is negative sign which is the typical situation.

Generic DFE



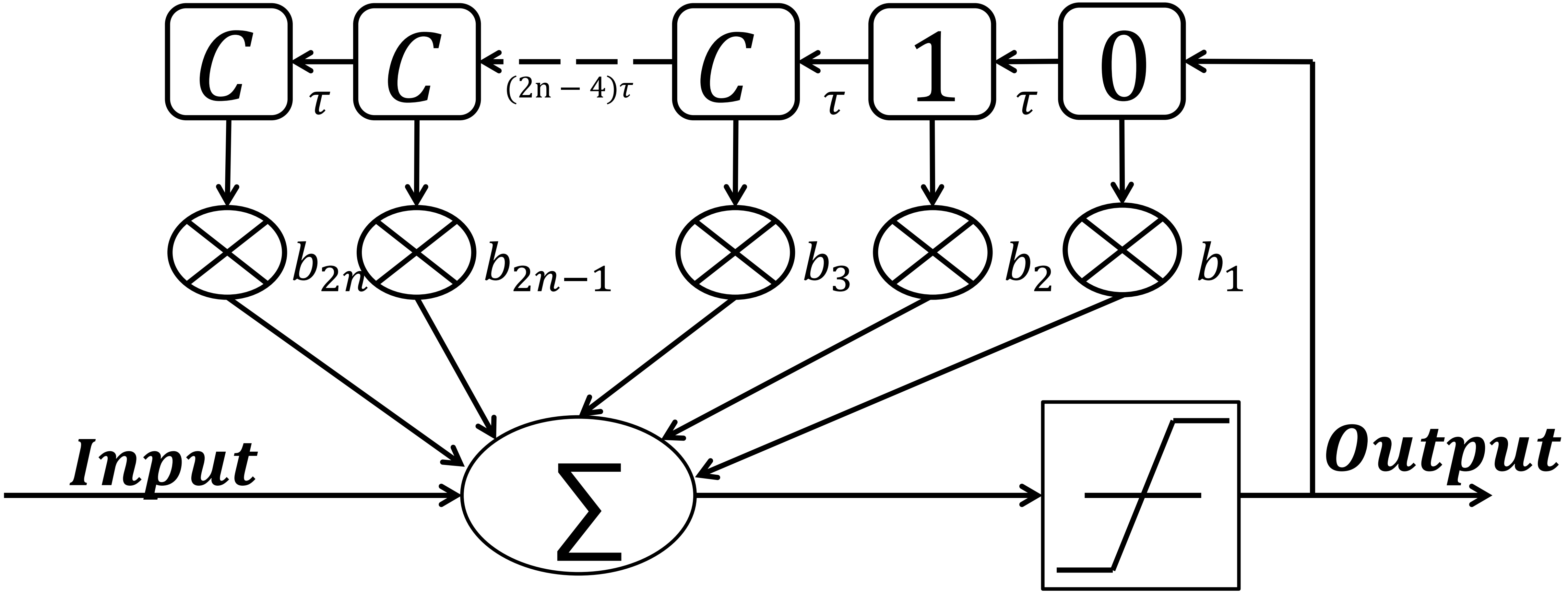
- C indicates a correct bit.
- The DFE taps are assumed to be set such that they remove the ISI caused by the applicable previous bit.

When the initial bit has errored from a “zero” to “one”



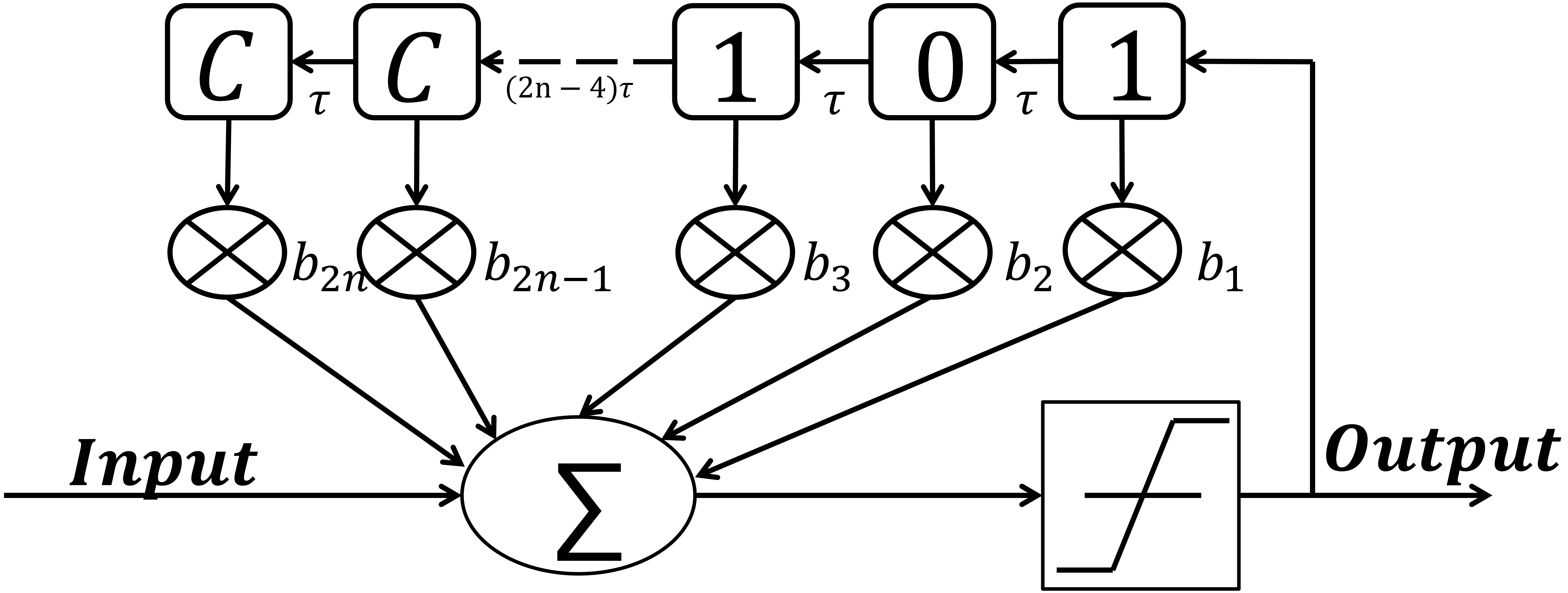
- With b1 negative if the next input bit is a zero the input to the slicer will be even more negative and the probability of an error becomes very low
- If the next input bit is a one then the signal into the slicer becomes smaller and the probability of it being detected as a “zero” becomes larger.
  - If b1 is -1 the input signal is at the normal “zero” level and the probability of it being detected as zero is almost 100%.
  - If b1 is -0.5 the input signal to the slicer will be at the slicer level and the probability of error is 50%.

When the error has propagated to the second bit.



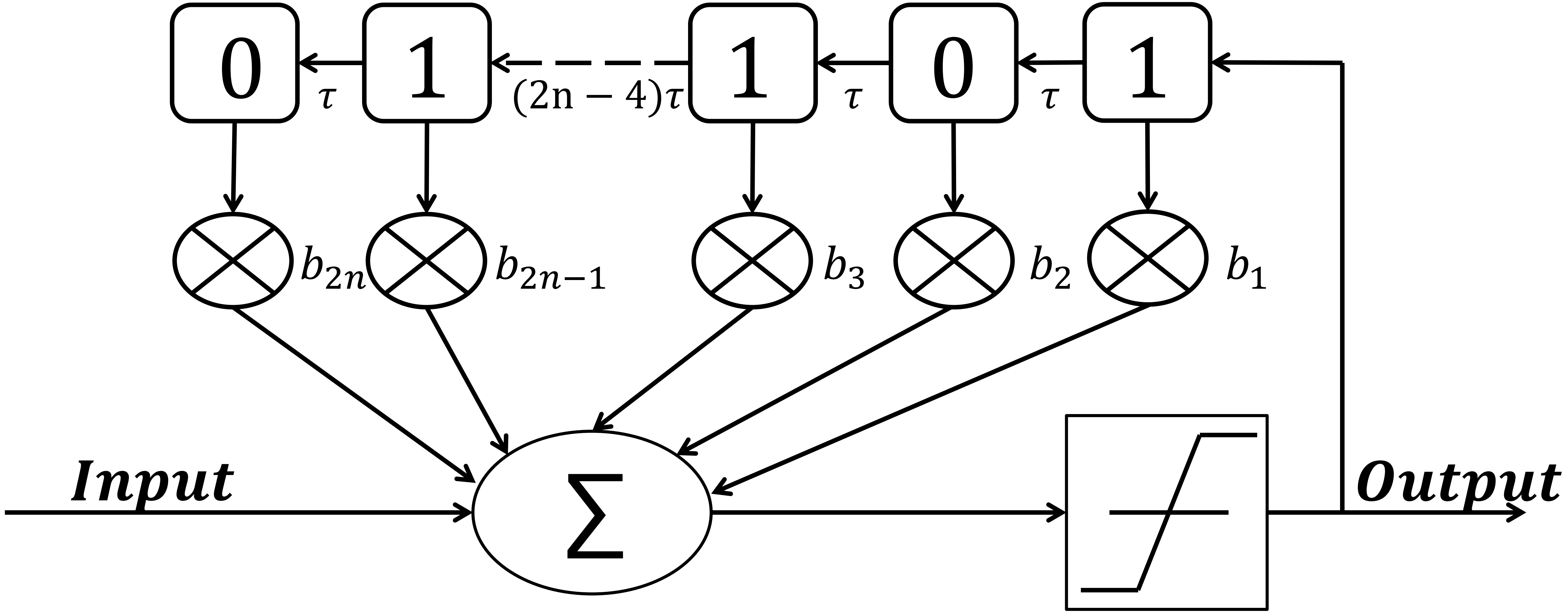
- With b1 negative the feedback from the first tap will be positive and the feedback from the second tap will depend on b2. If b2 is also negative then the feedback error will be smaller. If b2 is positive the feedback error will be larger.
    - The feedback error is b2-b1 and is likely to be positive. (It will be if b1 is negative and dominant.)
- If the next bit is a zero and b2-b1 is positive then the signal into the slicer becomes larger and the probability of error becomes larger.
    - If (b2- b1) is 1 then the probability of error is close to 100%.
    - If (b2- b1) is 0.5 the input signal to the slicer will be at the slicer level and the probability of error is 50%.
- If the next bit is a one and b2-b1 is positive the probability of error is very low.

When the error has propagated to the third bit.



- The error in the feedback is now equal to b1-b2+b3.
- If the next bit is a one and b1-b2+b3 is negative then the signal into the slicer becomes smaller and the probability of error becomes larger.
  - If (b1-b2+b3) is -1 then the probability of error is close to 100%.
  - If (b1-b2+b3) is -0.5 the input signal to the slicer will be at the slicer level and the probability of error is 50%.
- If the next bit is a zero and b1-b2+b3 is negative then the signal is even more negative and the probability of error is very low.

When errored bits have filled the shift register.



Analysis when errors have filled the shift register

- **The feedback amount is**

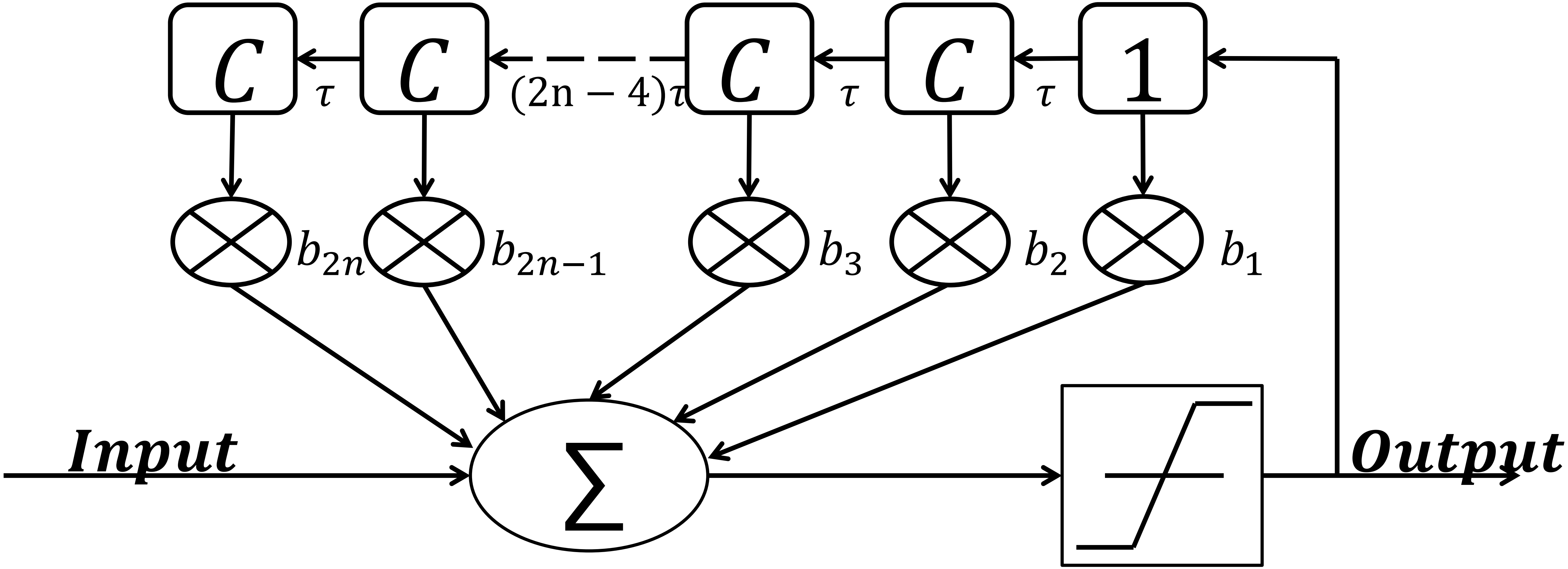
$$\sum_{m=1}^{n} b_{(2m-1)} - \sum_{m=1}^{n} b_{2m}$$

- **If long error bursts are the ones that create problems (eg with RS FEC) problems will only occur when the shift register is full of this errored bit pattern and it is better to ensure that the errors don’t propagate at a very high probability in this condition.**
- **The existing criterion of having Mod bn for all the taps of <0.35 or <0.5 does not do this. (with b1=-0.35, and b2=+0.35 even if all the other taps are zero there is a very high probability of very long error bursts.**
- **A criterion to limit error propagation in this situation is**

$$\sum_{m=1}^{n} b_{(2m-1)} - \sum_{m=1}^{n} b_{2m} > -0.5$$

Other considerations. If b1 is positive, and initial error is “Zero” mistaken as “one”

QLOGIC



- With b1 positive if the next input bit is a one the input to the slicer will be even more positive and the probability of an error becomes very low
- If the next input bit is a Zero then the signal into the slicer becomes larger and the probability of it being detected as another “one” becomes larger.
  - If b1 is 1 the input signal is at the normal “one” level and the probability of it being detected as one is almost 100%.
  - If b1 is 0.5 the input signal to the slicer will be at the slicer level and the probability of error is 50%.

Analysis when errors have filled the shift register with b1 positive.

- For the error propagation to occur the shift register will get filled with all "ones" (or all "zeros" if the original error had been a "one" mistaken as a "zero")
- The feedback amount will be

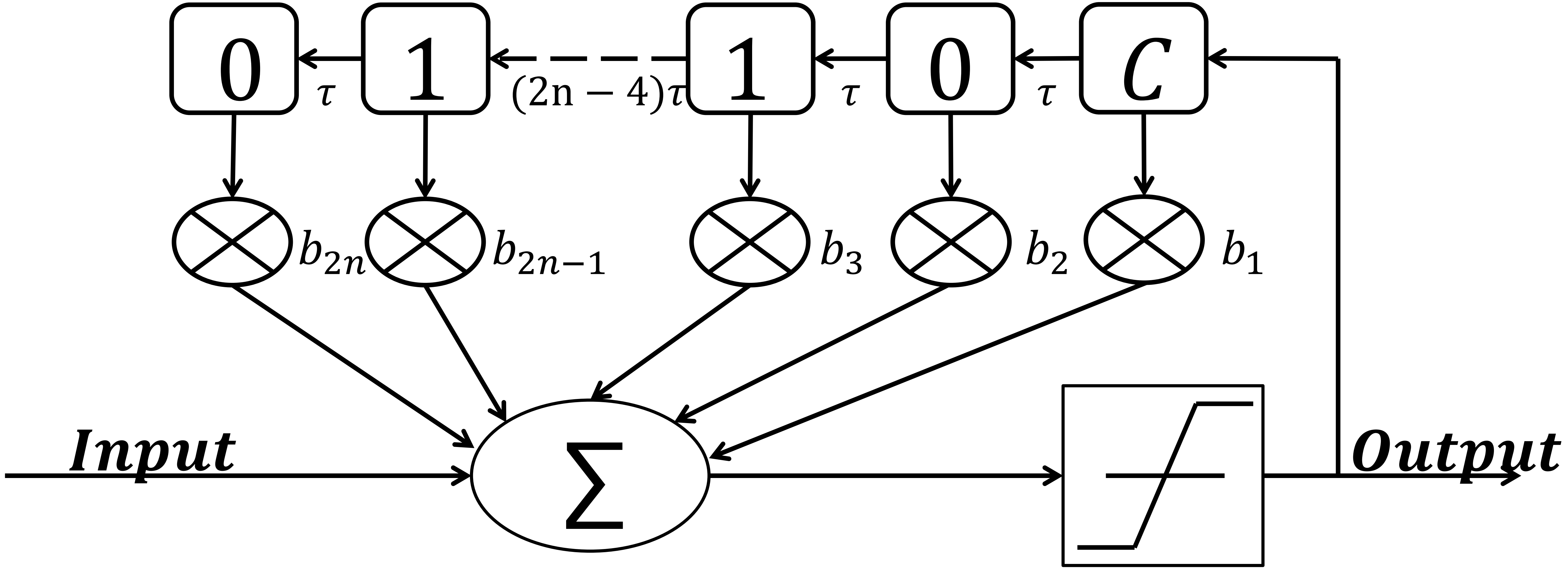
$$\sum_{m=1}^{2n} b_{(m)}$$

- The criterion to limit error propagation with long error bursts would be

$$\sum_{m=1}^{2n} b_{(m)} < 0.5$$

- This is however a very unlikely scenario and I doubt if it is worth complicating specifications by adding it.

Other considerations. If the input sequence is 010101010111010



- When the second adjacent one is input the correct decision will be made and the shift register will look as above.

Analysis of 0101010101111010

- The feedback error is now

$$\sum_{m=1}^{n-1} b_{(2m+1)} - \sum_{m=1}^{n} b_{2m}$$

- If this is -0.5 the probability of error is still 50%.   The error burst will restart. It will just have one correct bit in the middle of a block of errors.