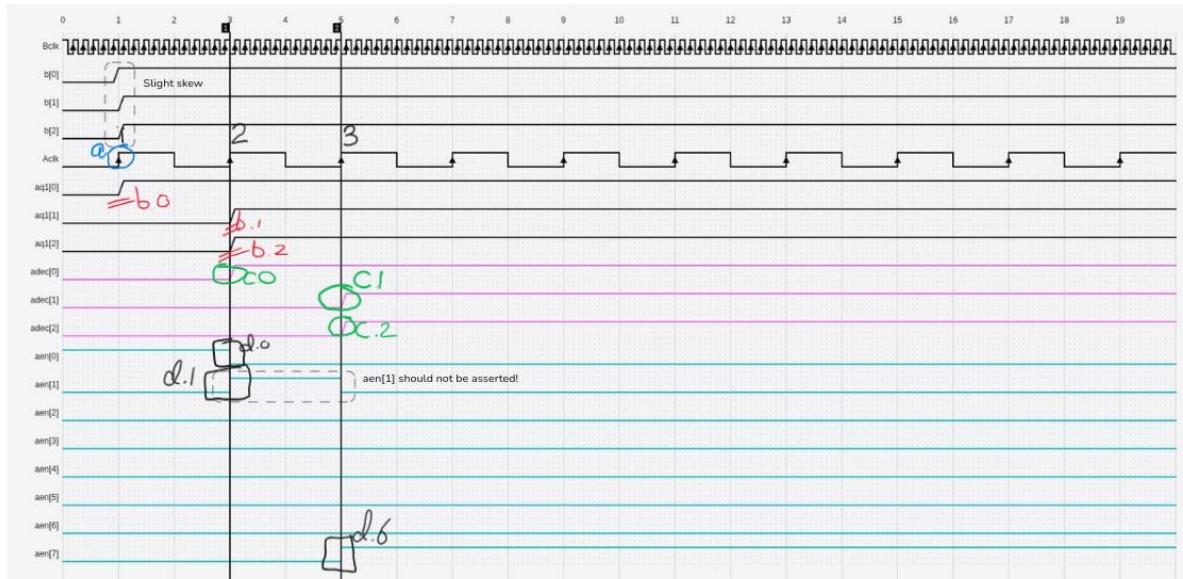


Q 1)

The 2FF design for combating metastability has a critical flaw: **it does not ensure data consistency, leading to uncertainty about which positive edge of which clock the signal will stabilise. The most significant issue is that some bits may reach the required stable value before others.** This discrepancy can cause the decoder to receive an incorrect combination of 0s and 1s, ultimately distorting the intended number during decoding. This problem is clearly illustrated in the waveform presented in the figure.

Q 2)



As we can see, the values of $b[0]$, $b[1]$, and $b[2]$ change simultaneously from 0 to 1 near the pos-edge 1 of $Aclk$, as labelled. We assume the change in signal occurs during the setup time of the first flip-flop layer. Due to input instability at setup time, the output tends to be metastable and can settle to 0 or 1 during the current clock cycle. This settling is not deterministic as stated above. If it tends to settle after the $Aclk$ clock cycle, the output will not be reflected.

For $aq1[0]$, it settles to 1 at the 1st pos-edge itself (as highlighted in $b0$), and the other two settle to 0 at the same point. But the 2nd pos-edge of the $Aclk$ detects the value of $b[1]$ and $b[2]$ to be 1 and hence is reflected the same in 2nd $Aclk$ (highlighted as $b1$ and $b2$). The 2nd flip-flop ensures that the final signals $b1$ and $b2$ are reflected in $c1$ and $c2$ on the 3rd $Aclk$ edge.

Now, when this encoded signal is fed into the decoder. During the 1st $Aclk$ pos-edge, it detects the sequence as 000($adec[2], adec[1], adec[0]$) as shown in $d0$. On the second clock edge, when the input signal from $b[0]$ has settled in $adec[0]$, but the other two have not, the decoder reads 001, and $aen[1]$ is high on that clock edge, as shown in $d1$. This timing difference in stabilising is what is causing this misinterpreted data to be sensed. But after the 3rd $Aclk$ edge, the decoder receives the correct signal (i.e., 111) and gives the correct output of $adec[7]$, as shown in $d6$.

Q 3)

The main mistake here is treating a multi-bit encoding sequence, which relies on the order of 0s and 1s, as if it were a sequence of independent bits, and using single-bit 2ff synchronizers for the CDC.

The decoder does not wait for all three signals to pass through the setup completely and is always accepting the encoded sequence. Each bit is synchronised separately; thus, during data processing in the 2ff, it picked up the intermediate values and decoded them, giving the wrong output for one clock cycle before the processed signals were presented at the 3rd clock cycle.