Questions :

1. Explain why synchronizing each bit of an encoded multi-bit control signal independently can lead to incorrect decoding in the receiving clock domain.

2. Using the timing diagram, describe how skew between b[1] and b[0] causes adec[2:0] to momentarily take an invalid intermediate value.

3. Identify the fundamental CDC design mistake illustrated in this figure.

4. Propose three different design techniques that can be used to safely transfer this control information across clock domains without generating spurious decoded outputs.

Answers :

1.Every bit of the three bit encoded signal can reach the input port of Flip Flop 1 in clock domain A at slight different times due to skew or some fundamental delay in the path. Apart from that all three synchronizers also might have some other flaws and delays which can cause inconsistent flow off all signals simultaneously. This can make decoding at receiver domain produce invalid outputs for some clock cycles.

2. As we can see in the waveform , b[0] changes from 0 to 1 before the clock of B domain comes and it might be possible that it changes to 1 before the setup time of clock B. But b[1] and b[2] change from 0 to 1 after that clock. This causes the flip flip 1 in receiver domain to capture b[0] one cycle prior to b[1] and b[2] which in turn indicates that B is changing from 000 -> 001(for one clock cycle) -> 111 thus adec[2:0] momentarily takes 001 as input , in one clock cycle , and gives 1 as an invalid intermediate output rather than directly going from 0 to 7.

3. The fundamental CDC mistake is that the input coming from the bclk domain are multi bit signals. So, we should never ideally pass multi bit signals bit by bit across 2 different clock domains.

4.

Method 1: Gray code encoding

Concept: Use Gray code where only one bit changes per transition.

When bits go across 2 clock domains, for multi bit systems, it is beneficial to use gray codes as they have only 1 bit change between 2 successive numbers so even if there is any error during transition, the resultant number would be either the same or a number ahead which is acceptable whereas in normal binary(BCD) system the number can change into anything which could cause false full or false empty conditions. (Eg in transition from 5 to 6 due to error it can go from 5 directly to 7 in BCD system which can create a false full condition).

Method 2: Use One hot encoding in B clock domain itself

Concept: Generate one-hot signals directly in bClk domain, synchronize each independently.

All bits become independent as they are already decoded in the source domain.There is no extra combinational logic required in the receiver domain so wont cause much skew or errors.

Method 3: Use Handshake Protocols

Concept: Use control signals to ensure data stability before sampling.

The source clock domain will send data only after all bits are synchronized correctly, and when it receives a ready signal from the receiver clock domain. SO chances of skew would be less as the receiver would request for data only when they become synchronized and are stable.