R/~W Ringing Problem on UF 68HC12 Board
(if Bad CE when Writing)

Write Cycle

ECLK

A7:0

Addr7:0

Multiplexed AD15:8

Addr15:8

Data7:0

R/~W

DBE

Latched Address (A15:8)

old latched address from previous cycle

new latched address from current cycle

Note: This problem exists if the write part of a RAM chip enable has an equation like this:

\[
\text{RAM\_CE} = \sim \text{RESET} \ast f(\text{Addresses}) \ast W, \text{ where } R/\sim W=R(H)=W(L)
\]

Note that there is no E-clock in the above equation.

Typical Example of the Problem (assumes RAM at SC000-CFFF):
1. Write $55$ to address $SC000$, then write $55$ to $SC001$, and finally one more write of $55$ to $SC002$.
2. You will see upon reading memory, $SC000$ has contents $SC0$, $SC001$ has contents $SC0$ and $SC002$ has contents $55$ (correct value).

Explanation of the Cause:
1. The RAM\_CE is true (using the above equation) in the first half of the E-clock, every time W is true (since the previous address is also within the RAM’s memory map). Since the previous cycle’s address (high byte) is on the address/data bus (A15:8/D7:0) during the early part of the first half of the E-clock when W is true, the high byte of the address is what gets written to the previous RAM address (over writing the correct value that had been previously stored).
2. The reason the last address gets the correct value is because the assumed next instruction is not another write cycle. If it was, it would also get changed to the wrong value.

A Solution for the Problem:
Add the E-clock signal to the write portion of all chip enable equations. Here is a short example. Let’s say we want to put an 8-bit output latch at $S8000$ and use partial-address decoding. The CE equation should be:

\[
\text{CE\_for\_Write} = \sim \text{RESET} \ast A15 \ast \sim A14 \ast \sim A13 \ast \sim A12H \ast W \ast E
\]

This will ensure that the chip enable will only go true when E is high thus avoiding the ringing area altogether.