What's Wrong with this?
Some questions and a sum
- with acknowledgements to Ran Ginosar

How does it work?
- $T$ is set for one cycle
- Receiver has at least 2 chances to read $T$
- What can go wrong?

- In a next generation chip, the sender is faster
- Reuse limitation

- Used in Multi-Sync cases
- New data sent every cycle
- This one is extremely dangerous (MTBF=0)!
- Why?

- Some bits may go metastable
- Some will resolve to 0, others to 1
- Guaranteed to fail…

- What's wrong with this one?
A Two-Latch Synchronizer

- The synchronizers were left out...

---

Conservative Synchronizer

- The universe is only $10^{10}$ years old...

Now I am really safe!

MTBF(T) = $\frac{\tau}{T_w} F_C F_D F_R$

MTBF(2T) = $\frac{50 \text{ psec}}{200 \text{ MHz} \times 200 \text{ MHz}} = 2 \times 10^9 = 10^{10}$ years

Conservative Synchronizer

- What’s wrong with being conservative?
  - This guy doesn’t know what he’s doing...
  - So I am worried about OTHER synchronizers that he has done!

---

What would the latency be if...?

- Imagine a future GALs SoC with 100 independent unpredictable clock zones, $f_c = f_d$
  - 5GHz, $T_w = \tau = 10$ps
- How much time do 100 simple synchronizers need to get an overall reliability of 1 year?
  - How many clock cycles? (40c, or 2 cycles)
- How long does it take to transmit data from one side of the chip to the other? (across 9 boundaries) (9x 400 ps, or 1.8ns)