Synchronizers and Arbiters
OR
1000 years of indecision

David Kinniment
University of Newcastle

What's the problem?

- Measuring continuous quantities (like time and voltage) is not precise.
- Comparing two times is hard
  - 2.153 nanoseconds
  - 2.152 nanoseconds
- To know which is the greater you need to compare the digits until you find a difference
- This might take for ever.
- Comparing two integers is easy
  - 23
  - 23

Comparison Hardware

- Digital comparison hardware (which compares integers) is
  - Fast
  - Bounded time
  - Result can be <, >, or =
- Analog comparison hardware (which compares reals) is
  - Normally fast, but takes longer as the difference becomes smaller
  - Can take forever
  - No such result as =
- Synchronization and arbitration involve comparison of reals

History & Philosophy

- Abu Hamid Ibn Muhammad Ibn Muhammad al-Tusi al-Shafi'i al-Ghazali ~1100
  - "Suppose two similar dates in front of a man who has a strong desire for them, but who is unable to take them both. Surely he will take one of them through a quality in him, the nature of which is to differentiate between two similar things"
  - He felt that this demonstrated free will
- Jehan Buridan, Rector of Paris University ~1340
  - Buridan's Ass (A dog with two bowls?)
  - "Should two courses be judged equal, then the will cannot break the deadlock, all it can do is to suspend judgment until the circumstances change, and the right course of action is clear"
  - He's not so sure

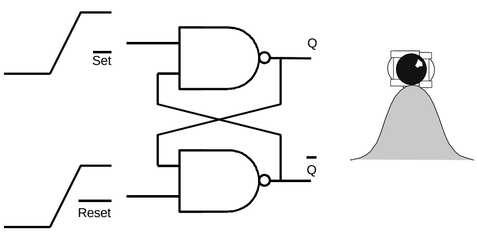
Digital Computers

- Voltages have a finite number of values in a computer, 1 and 0
- Time has a discrete number of instants in a synchronous system

BUT

- Computers have to talk to other computers and to people who are not synchronous
- Ivor Catt 1966
- Chaney and Littlefield 1969/72

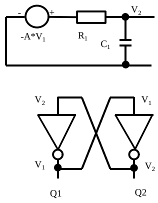
Metastability is....



Not being able to decide…

Linear Model

- Simple linear model leads to two exponentials
- t_a is convergent, t_b is divergent



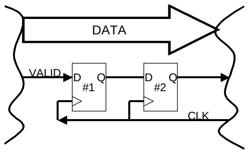
$$t_1 = \frac{C_1 . R_1}{A} , t_2 = \frac{C_2 . R_2}{A}$$

$$0 = t_1 . t_2 . \frac{d^2 V_1}{dt^2} + \frac{(t_1 + t_2)}{A} . \frac{dV_1}{dt} + (\frac{1}{A^2} - 1).V_1$$

$$V_1 = K_a . e^{\frac{-t}{t_a}} + K_b . e^{\frac{t}{t_b}}$$

The synchronizer

- Clock and valid can happen very close together
- Flip Flop #1 gets caught in metastability
- We wait until it is resolved (1 Clock cycle)



How often does it fail?

- Suppose the clock frequency is f_c, and the data rate f_d.
- In M seconds we have M.f_c clocks.
- The probability of a data change within t_D of any clock is $t_D.f_d$, so there will be one within M seconds if $t_\Delta = \frac{1}{M . f_c . f_d}$
- The time taken to resolve this event is $\frac{T_w}{t_\Delta} = e^{\frac{t}{t}}$ (T_w is a constant)

MTBF

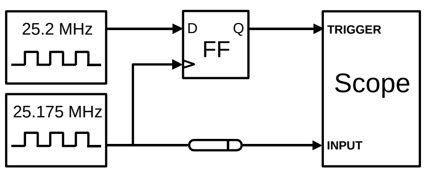
$$MTBF = \frac{e^{t/t}}{T_w . f_c . f_d}$$

- For a 0.18µ process t is 20 – 50 ps
- T_w is similar
- Suppose the clock frequencies are 2 GHz
- t needs to be > 25 t (more than one clock period) to get MTBF > 28 days

State of the art

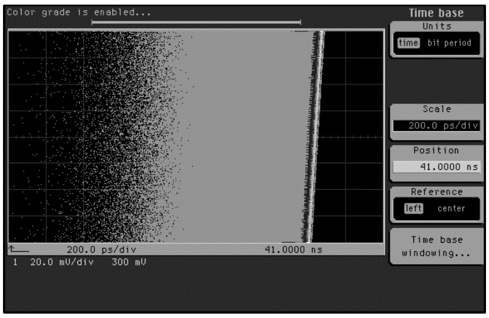
- You require about 35 τ s in order to get the MTBF out to about 1 century. (That's for 1 synchronizer)
- Each typical static gate delay is equivalent to about 5 τ s in a properly designed synchronizing flop. (You can increase V_{dd} on the flop to get it faster)
- You must assume a 'malicious' input to the synchronizer. Nevertheless, this only adds about 5 τs to the delay.

Metastability Measurement



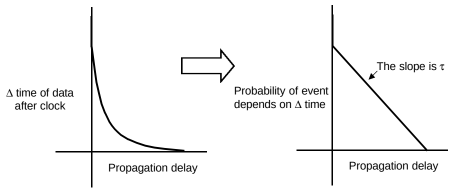
UK Forum

Metastability



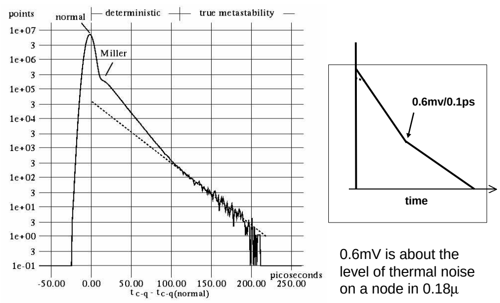
UK Forum June 30 2003 13

Event Histogram



UK Forum June 30 2003 14

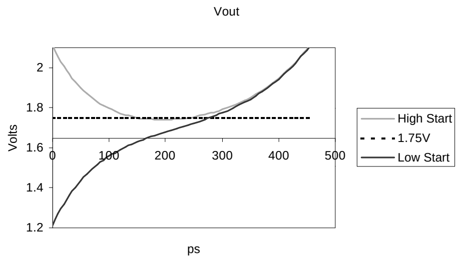
Measured Histogram



0.6mV is about the level of thermal noise on a node in 0.18µ

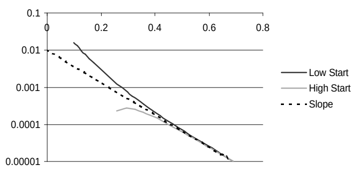
UK Forum June 30 2003 15

Why isn't it straight?



UK Forum June 30 2003 16

Histogram of events

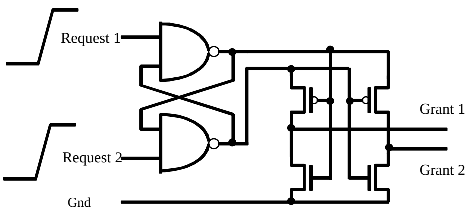


- Probability of an event occurring within 10ps of a particular output time

UK Forum June 30 2003 17

The arbiter (MUTEX)

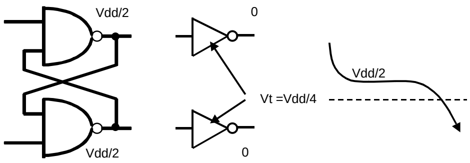
- Asynchronous arbitration
- No time bound



UK Forum June 30 2003 18

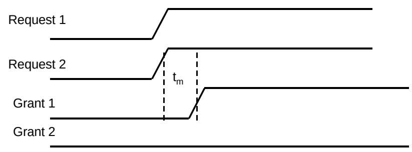
Metastability filters

- Low (or high) threshold inverters
- Measure divergence
- These define the time to reach a stable state

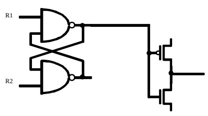


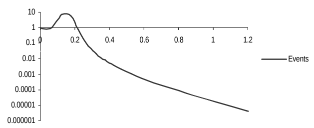
Arbitration time

- Unlike a synchronizer, an arbiter may take for ever.
- It usually doesn't, long responses are rare.
- On average the time is only τ longer than the normal response.
- Outputs are always monotonic



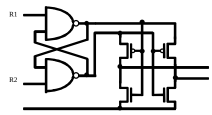
MUTEX with low threshold output

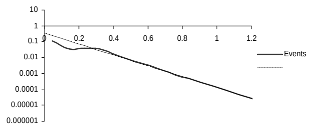




- Starts high, needs to go low to give output
- Threshold about 100 mV low

MUTEX with filter

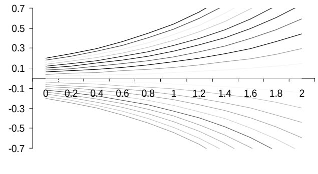




- Needs more than 1V difference to give output
- Slower

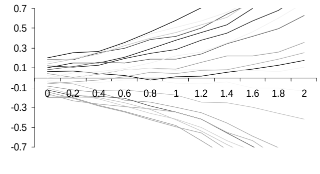
Does noise affect *t* ?

- Probability of escape from metastability does not change with gaussian noise (Couranz and Wann 1975)



Does noise affect *t* ?

- Probability of escape from metastability does not change with gaussian noise (Couranz and Wann 1975)

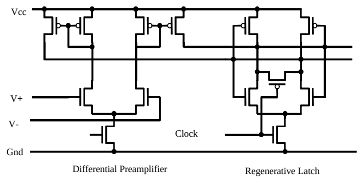


4

UK Forum

Measuring noise

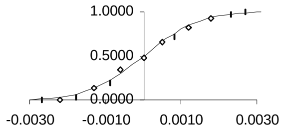
- Slowly change bias to latch
- Keep clocking and observe output



Noise measurement

Probablility of an output 1 as a function of input voltage difference

Our measurement of approximately 1.7mV RMS at the input corresponds to about 0.6mV total between latch nodes

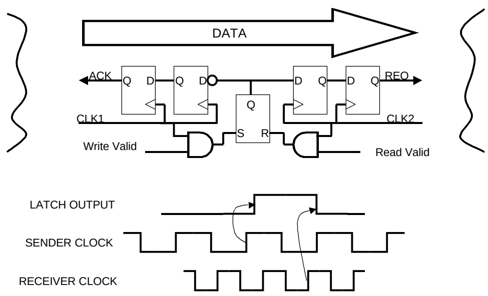


$$\sqrt{\frac{4kT}{C}} \approx 0.7mV$$

Noise

- Doesn’t affect statistical probability of long metastaibility times.
  - If input time distributions are constant
  - If they’re bad it can make things better
- Means GALS systems with many synchronizers are indeterminate.
- Nanometre systems are more affected by noise because C and V are smaller

Single latch synchronizer

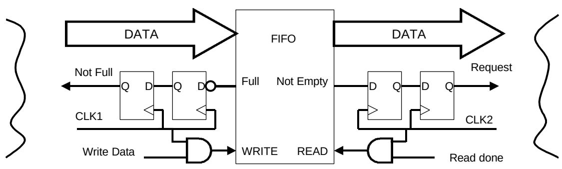


latency

- It takes one - two receive clocks to synchronise the request
- Then one – two write clocks to acknowledge it
- Significant latency (1-3 clocks)
- Poor data rate (2 – 6 Clocks)

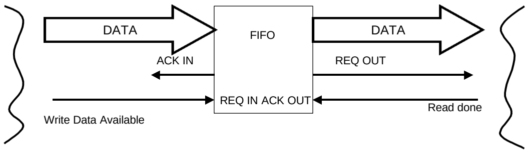
FIFO

- Can improve data rate by using a FIFO
- But not latency (which gets worse)
- FIFO is asynchronous (usually RAM + read and write pointers)



Don't synchronise when you don't need to

- If the two clocks are locked together, you don't need a synchroniser, just an asynchronous FIFO big enough to accommodate any jitter/skew
- FIFO must never overflow

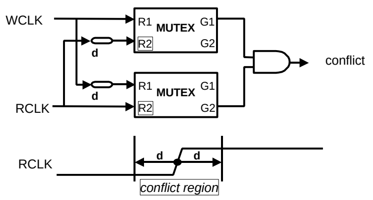


Timing regions can have predictable relationships

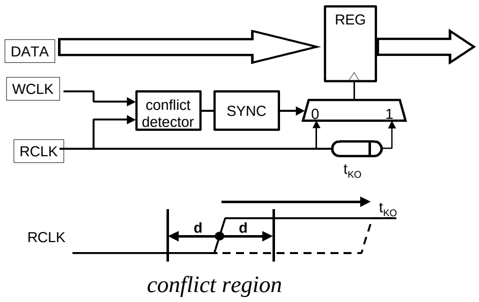
- Locked
  - Two clocks from same source
  - Linked by PLL
  - One produced by dividing the other
  - Some asynchronous systems
  - Some GALS
- Not locked together but predictable
  - Two clocks same frequency, but different oscillators.
  - As above, same frequency ratio

Pre synchronisation

- Predict when clocks are going to conflict and delay synchronization
- Dike's conflict detector



Clock delay synchronizer



Pre synchronizer latency

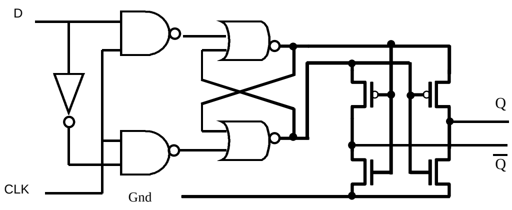
- Nominally 0 – 1 clock cycle
- Relies on accurately predicting conflicts
- Clocks must remain stable over synchronisation time.
- Always lose t_{ko} of next computation stage
- Alternative: shift all conflicts to next read cycle
  - On average this loses 2d
  - 2d must be big enough to cover any clock drift/jitter over synchronization time

Post synchronisation

- Clocks may not be easy to predict
  - Increased jitter
  - Cross talk
  - Drift over synchronization time
- Asynchronous to synchronous
- Go ahead anyway, and recover if there was a conflict

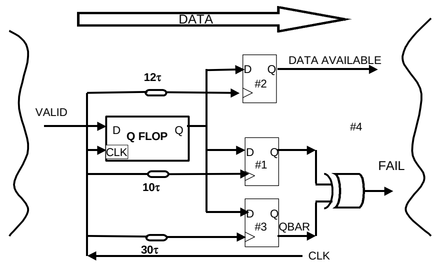
Q Flop

- With CLK low, both outputs are low
- With CLK high, Q becomes equal to D after metastability



Was it OK?

- FF#1 is set after 10τ, FF#2 after 12τ, FF#3 much longer, say 30τ
- Latency is normally 12τ, but synchroniser fails often



When to recover

10τ	12τ	30τ	Fail	Remark
0	0	0	0	No data available
0	0	Metastable	?	Unrecoverable error
0	Metastable	1	1	Change between 10τ and 30τ, return to original state
Metastable	1	1	0	The partially synchronized transition must have occurred at 10τ, so the 12τ signal Read Data available will be high
1	1	1	0	Normal data Transfer

Post Synchronisation latency

- Recovery means restoring any corrupted registers, and may take some time,

BUT

- Probability of recovery operation is e^{-10}, so little time lost on average.
- Average synchronization time
  0 –1 cycle + 12τ + Recovery Time . e^{-10}

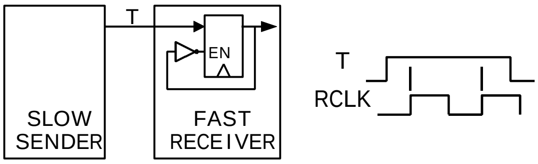
Conclusions

- Synchronization/arbitration requires special circuit elements
- They're not digital!
- If there's a real choice, and bounded time you will have failures.
- The MTBF can easily be made longer than the life of the universe
- Latency is a problem, but not insuperable.
- Synchronizers are not deterministic.

What's Wrong with This? Some questions and a sum

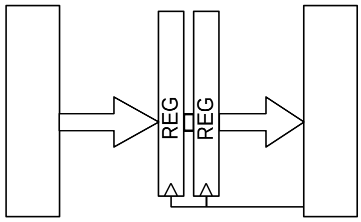
- with acknowledgements to Ran Ginosar

Slow-to-Fast Synchronizer



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

Parallel Synchronizer



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

A Two-Latch Synchronizer

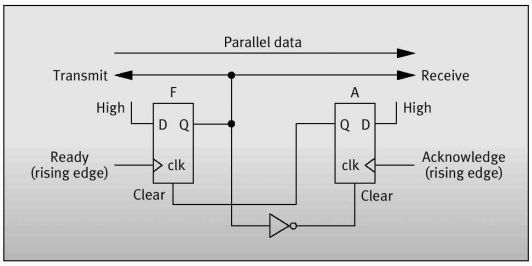
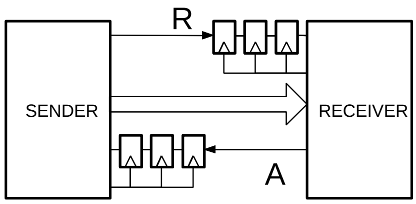


Figure 1: Parallel passing is normally used as the traditional method to generate a flag.

- What's wrong with this one?

Conservative Synchronizer



- FC = 200MHz
- FD = FC
- T_W = 50 psec (0.18μ)
- τ = 10 psec

- What's the MTBF for
  - Two-FF sync ?
  - Three-FF sync ?
  - Eight-FF sync ?

Conservative Synchronizer

- F_C = 200MHz
- F_D = F_C
- T_W = 50 psec (0.18μ)
- τ = 10 psec

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

*Now I am **really** safe!*

$$MTBF(T) = \frac{e^{T/t}}{T_W \cdot F_C \cdot F_D} = \ldots 200MHz$$

$$= \frac{e^{5\ldots}}{50\,p\sec \cdot 200MHz \cdot 200MHz} = \ldots = 10^{203}\ years$$

$$MTBF(2T) = \frac{e^{10n\sec/10p\sec}}{50 \ldots 200MHz \cdot 200MHz} = \frac{e^{1000}}{2 \cdot 10^6} = 10^{420}\ years$$

What would the latency be if…?

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