EEE8068 Real Time Computer Systems
Synchronisation problem

Dr. Bystrov

School of Electrical Electronic and Computer Engineering

University of Newcastle upon Tyne

Aims and objectives

Aim: the problem of decision making
Objectives:

- History
- Formal definition and a solution
- Examples
- Application to Embedded Systems
- Analysis and the home task
- Conclusions

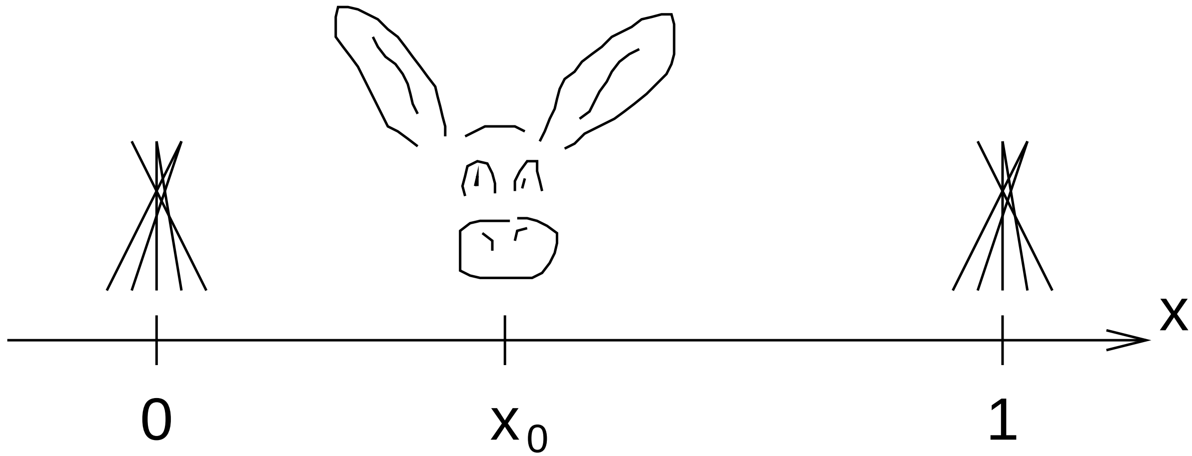
“Buridan’s ass” problem

Leslie Lamport: “Buridan’s Principle”, DEC, Systems Research Center, 21 Jan. 1086. http://citeseer.nj.nec.com/389583.html

- Jean Buridan, 14th century French philosopher.

An ass placed equidistant between 2 bales of hay (or some other food) starves to death!!! Why?

Lamport's model formulation



$0 \leq x_0 \leq 1$

The ass moves on its will;

time $t \geq 0$: position $A_t(x_0)$.

Assume that when the ass reaches a bale of hay it stays there forever.

$A_t(0) = 0$; $A_t(1) = 1$;

$0 \leq A_t(x_0) \leq 1$ if $0 < x_0 < 1$.

$A_t(x_0)$ is a continuous function of x_0.

Lamport's model analysis

- Since $A_t(0) = 0$ and $A_t(1) = 1$, by continuity there must be a finite range of values of x_0 for which $0 < A_t(x_0) < 1$.
  - These values represent initial positions of the ass for which it does not reach either bale of hay within t seconds.
- Such a range of values of x_0 exists for any time t, including times large enough to insure that the ass has starved to death by then.

Buridan's principle: A <u>discrete</u> decision based upon an input having a <u>continuous</u> range of values cannot be made within a <u>bounded</u> length of time.

“Asses” in the cars

A car at an unguarded railroad crossing, a donkey is driving. . .

- The car stops at the entrance.
- The driver must make a discrete decision:
  - to wait or
  - to cross.
- The decision must be made in the bounded length of time,
  - which by Buridan’s Principle is impossible.
- “Escapes” from the Buridan’s Principle:
  - take some predefined action in case of indecision;
  - introduce noise or randomness in the system.

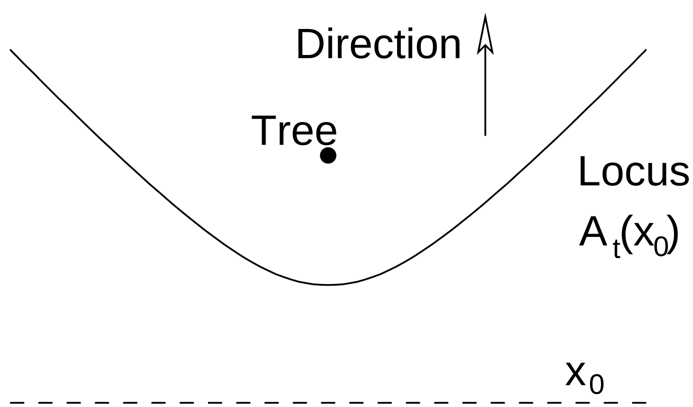
Modelling a car driver

- $A_t(x_0)$ – the position of the car at time t.
  - x_0 – the time at which the car reaches the entrance.
- The train reaches the crossing at time 0.
- If $x_0^{early} \ll 0$, then the car will cross before the train arrives, so $A_0\left(x_0^{early}\right)$ is on the far side of the track.
- If $x_0^{late} > 0$, then the driver will wait, hence $A_0\left(x_0^{late}\right)$ is in the near side of the track.
- By continuity there must be some time $x_0 : \; x_0^{early} < x_0 < x_0^{late}$, such that $A_0(x_0)$ is in the middle of the track!
- Moreover, the track has a finite width. . .

A tree in the middle of a motorway

Tree avoidance

- A discrete decision is possible in an unbounded time.
- Give the driver an option to slow down before hitting the tree (speed is a function of the distance to the tree).
  - Read the Highway Code, it is all about it!
- Then we are back to the original Buridan’s ass problem.
  - The ass driver will starve to death or at least cause a severe traffic jam.



More examples of the Buridan's problem

- Alice an Bob walking towards each other and deciding whether to pass to the left or to the right.
- Choosing between vanilla and chocolate ice cream.
- Pilot asses
  - the third dimension complicates the analyses, but the result is the same.
  - Can aeroplanes slow down or stop in flight?...
- Decision elimination is the only way to avoid starvation.
  - Alice and Bob may go to the church and marry...
  - One can buy half-vanilla half-chocolate ice cream...
  - One can avoid flying...
  - Air traffic controllers may lay the routes so there were no intersections...

Interrupt handling in computers

- Interaction between a computer and a peripheral device.
  - The device raises a flag (interrupt request set).
  - The computer checks the flag on execution of every instruction.
  - When the computer sees this flag, it interrupts normal processing and executes a special program to service the device.
  - The computer drops the flag (resets the request).
- Flag setting is **not** synchronised with the computer.
- The flag state is a continuous function of time and the computer **attempts** to make a binary decision in a bounded time.

An “ass” in the computer

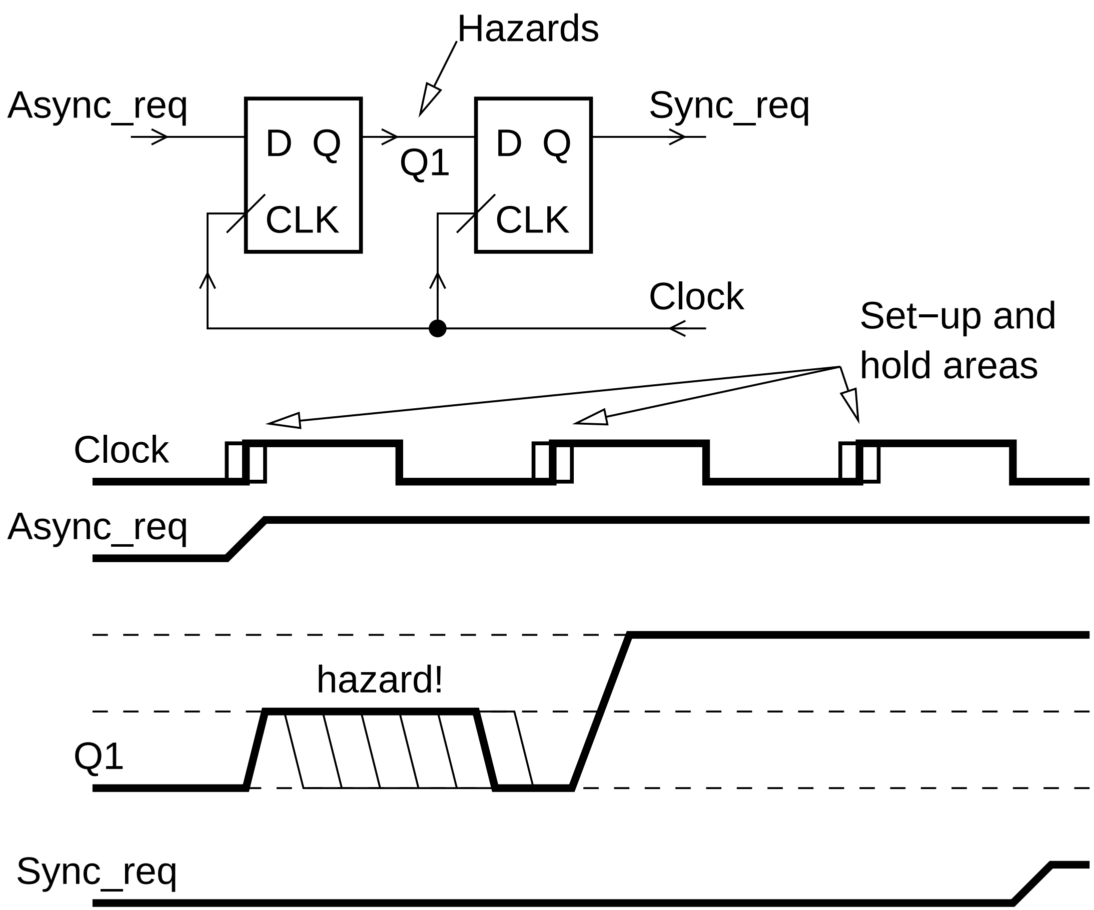
- Interrupt handling mechanism.
- It attempts to do the impossible;
  - remember the railroad crossing!
- Indecision manifestation – metastability;
  - bad voltages propagated;
    - $V_{cc} = 5V$, $V_{meta} = 2...3V$;
    - possibility of oscillations;
  - a bad voltage can be interpreted as 0 by some circuits and as 1 by the others.
- Arbiter Problem;
  - more time – less probability of a hazard;
  - unbounded time – asynchronous arbiters.

Buridan's Law of Measurement

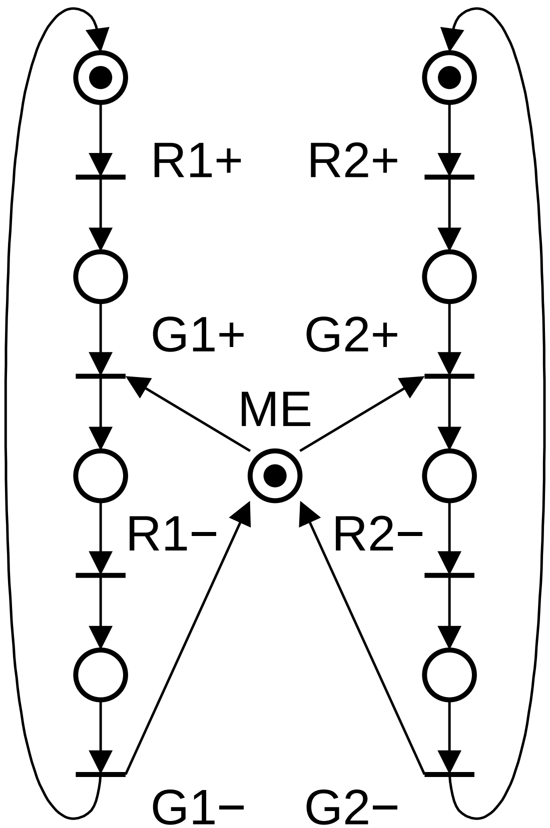
- If $x < y < z$, then any measurement performed in a bounded length of time that has
- a non-zero probability of yielding a value in a neighbourhood of x and
- a non-zero probability of yielding a value in a neighbourhood of z
- must also have a non-zero probability of yielding a value in a neighbourhood of y.

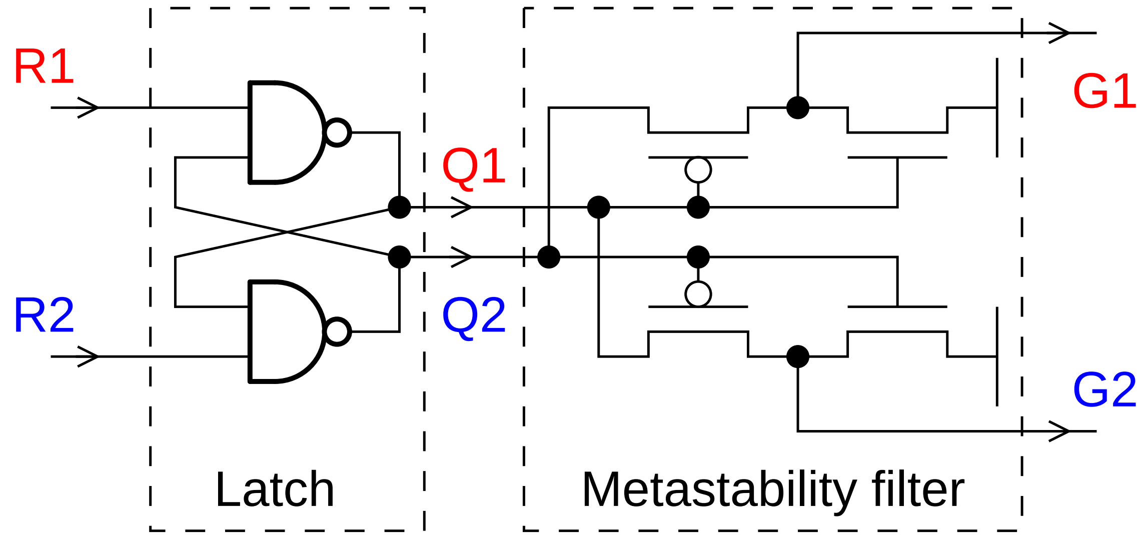
Synchroniser

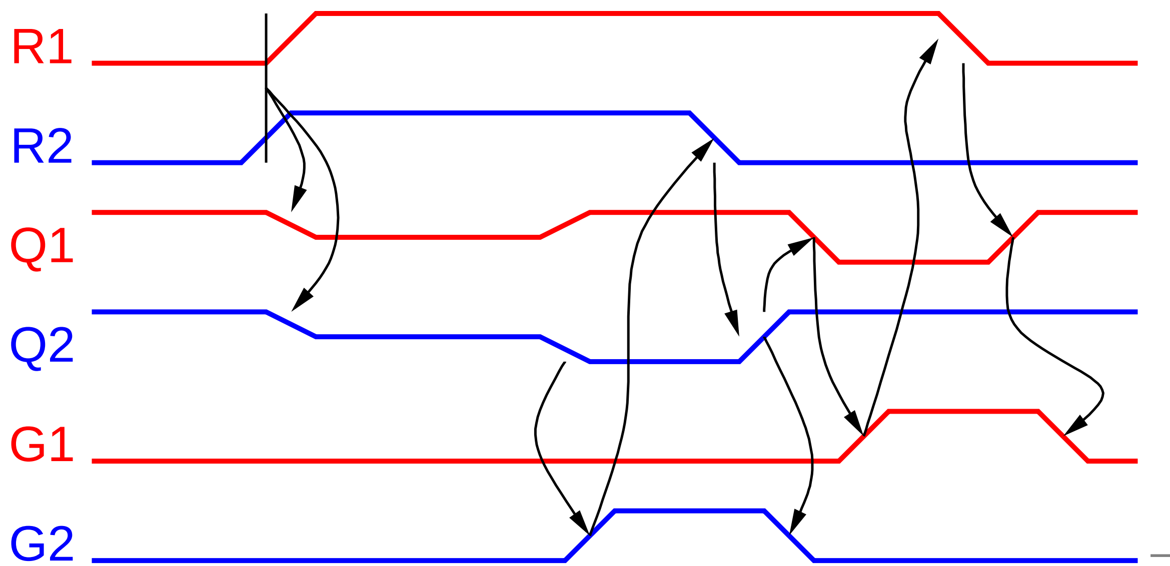
The “ass” is still there, but it is allowed more time to make a decision.



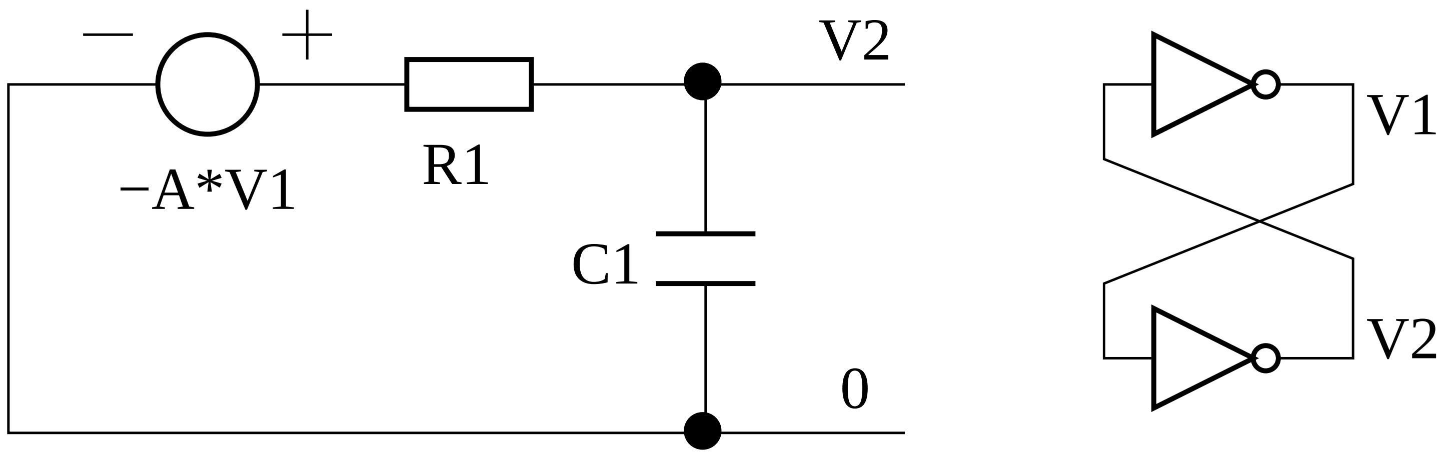
Asynchronous arbiter







Metastability modelling



- Small signal analysis – the area around the threshold of decision making
- Non-linear gates – linear approximation – linear ODEs
- Use Ohm's law, Kirchhoff's law (the sum of currents into a node = 0), $I_{cap} = C\frac{dV_{cap}(t)}{dt}$.

Kirchhoff's Law – ODE system

$$\begin{cases} C_1\frac{dV_2}{dt} + \frac{AV_1+V_2}{R_1} = 0 \\ C_2\frac{dV_1}{dt} + \frac{AV_2+V_1}{R_2} = 0 \end{cases} ;\ \tau_i = \frac{C_i R_i}{A_i};\ A_1 = A_2 = A;$$

$$\begin{cases} A\tau_1\frac{dV_2}{dt} + V_2 + AV_1 = 0 \\ A\tau_2\frac{dV_1}{dt} + V_1 + AV_2 = 0 \end{cases}$$

Solving the ODE system

$$\begin{cases} A\tau_1 \frac{dV_2}{dt} + V_2 + AV_1 = 0 \\ A\tau_2 \frac{d^2V_1}{dt^2} + \frac{dV_1}{dt} + A\frac{dV_2}{dt} = 0 \end{cases}; \begin{cases} \tau_1 \frac{dV_2}{dt} + \frac{V_2}{A} + V_1 = 0 \\ -\tau_1\tau_2 \frac{d^2V_1}{dt^2} - \frac{\tau_1}{A}\frac{dV_1}{dt} = \tau_1\frac{dV_2}{dt} \end{cases};$$

$$\tau_1\tau_2\frac{d^2V_1}{dt^2} + \frac{\tau_1}{A}\frac{dV_1}{dt} - \frac{V_2}{A} - V_1 = 0;$$

$$\tau_1\tau_2\frac{d^2V_1}{dt^2} + \frac{\tau_1}{A}\frac{dV_1}{dt} + \frac{A\tau_2\frac{dV_1}{dt} + V_1}{A^2} - V_1 = 0;$$

$$\tau_1\tau_2\frac{d^2V_1}{dt^2} + \frac{\tau_1}{A}\frac{dV_1}{dt} + \frac{\tau_2}{A}\frac{dV_1}{dt} + \frac{V_1}{A^2} - V_1 = 0;$$

$$\tau_1\tau_2\frac{d^2V_1}{dt^2} + \frac{(\tau_1+\tau_2)}{A}\frac{dV_1}{dt} + (\frac{1}{A^2} - 1)V_1 = 0;$$

The solution of the last ODE:

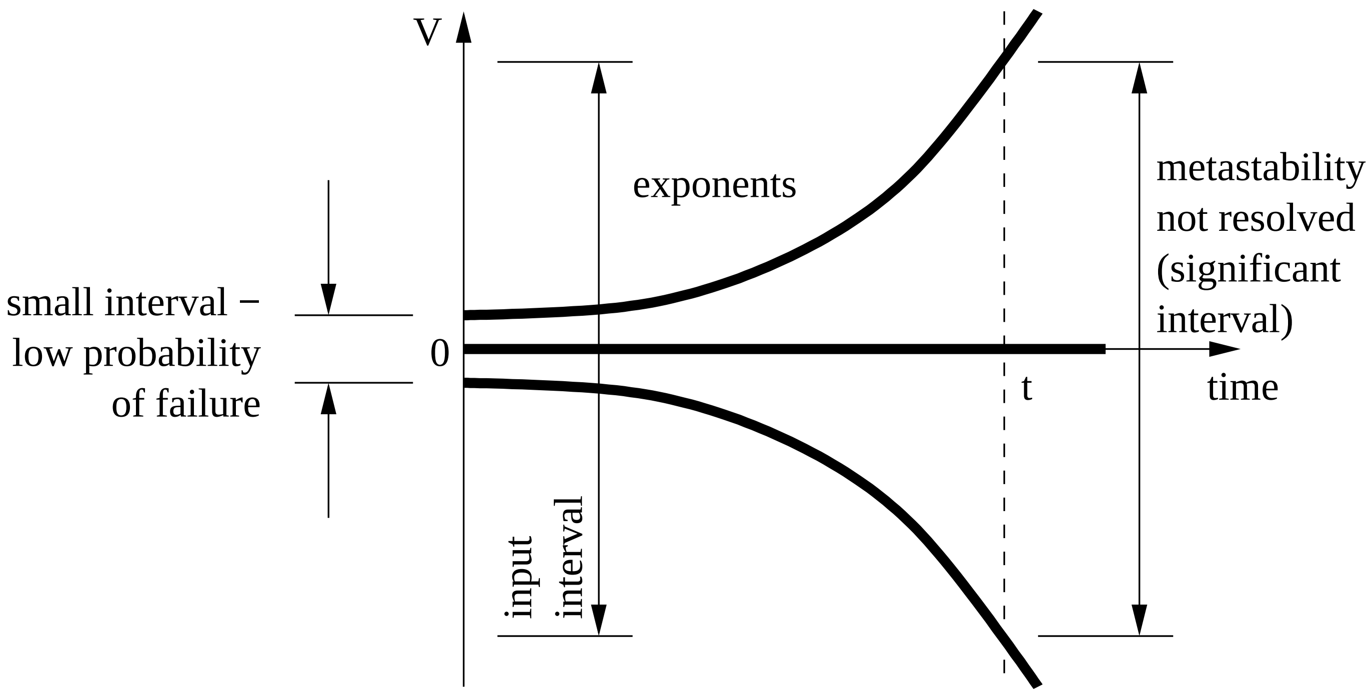
$V_1 = K_a e^{\frac{-t}{\tau_a}} + K_b e^{\frac{t}{\tau_b}}$;

V_2 is similar

K_a, K_b – initial conditions

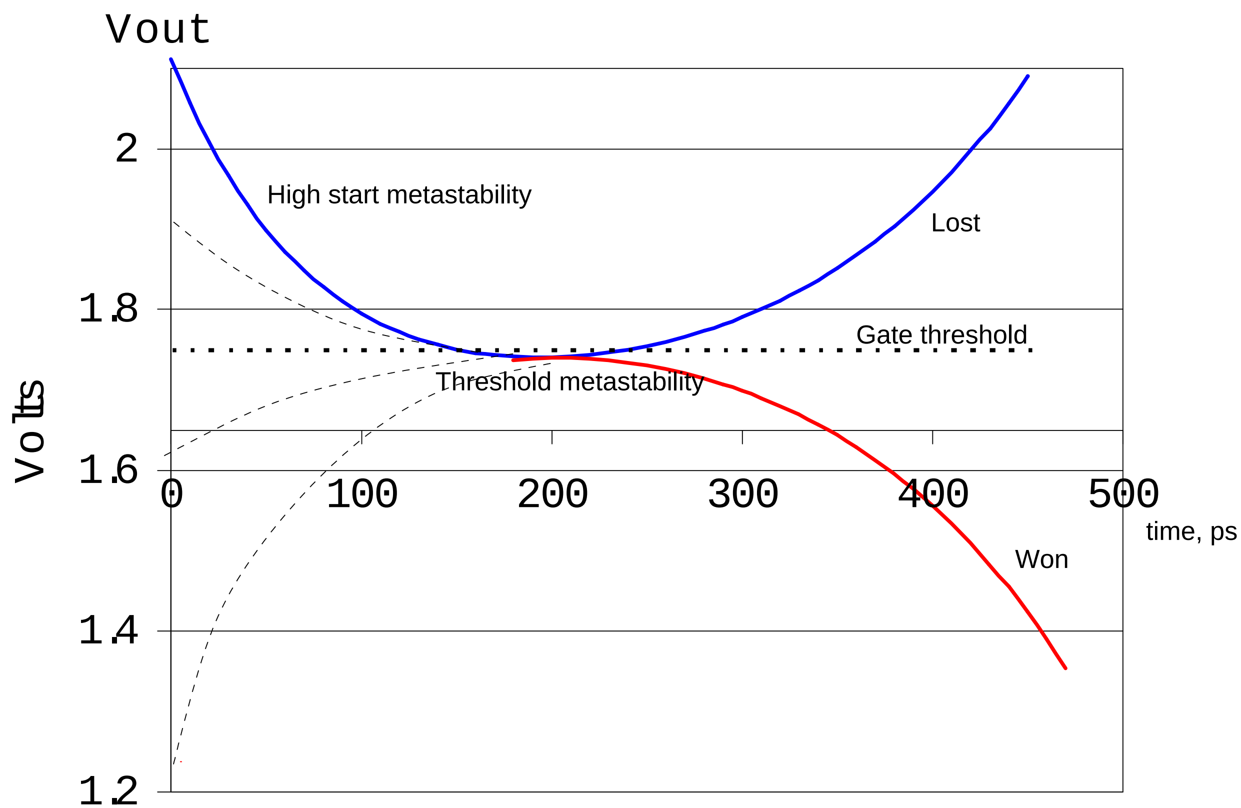
Probability of long metastability

$V_1 = K_a e^{\frac{-t}{\tau_a}} + K_b e^{\frac{t}{\tau_b}}$ – Exponent!



Assume, the input is uniformly distributed in a large interval. The interval resulting in not resolving metastability after t is small. Hence, the probability of long metastability is very low. It is <u>rapidly</u> decreasing with t. $MTBF \sim e^{\frac{t}{\tau}}$

Metastability – real circuit



- One channel (Q1), in one case loses the arbitration (going up), in the other case wins (going down)
- The imbalance exists at 0ps, but becomes visible only after 200ps

Metastability modelling (home task)

- Read **http://www.staff.ncl.ac.uk/david.kinniment/Research/papers/Jssc2002.PDF**
- Write a system of ODE for the above case
- Solve the system
- Draw a family of curves corresponding to the solution. Each pair of curves will correspond to a particular set of initial conditions.

Conclusions

Help me!

- ______________________________
- ______________________________
- ______________________________
- ______________________________
- ______________________________
- ______________________________

Well done!

Literature on arbiters/synchronisers

- T. J. Chaney, C. E. Molnar: “Anomalous behaviour of synchronizer and arbiter circuits”, IEEE Trans. on Comput., vol. C-22, 1973, pp. 421–422.
- **Synchronization and Arbitration in Digital Systems by David J. Kinniment (Hardcover - 7 Dec 2007)**
- **D. J. Kinniment, A. Bystrov, A. Yakovlev: “Synchronization circuit performance”, IEEE Journal of Solid-State Circuits, vol. 37, 2002, pp. 202–209. http://www.staff.ncl.ac.uk/david.kinniment/Research/papers/Jssc2002.PDF**
- R. Ginosar: “Fourteen ways to fool your synchronizer”, 9th IEEE Int. Symposium on Asynchronous circuits and Systems, Vancouver, BC, Canada, 12-15 May, 2003, pp. 89–96.