ACMs – a single bit

EE2807

Dr Fei Xia and Dr Alex Bystrov

Pools

- Writer and reader may access the ACM fully asynchronously
  - No synchronization between the two processes
  - No waiting by either side
  - Overwriting
  - Re-reading
- Conceptual single-space buffer
  - Permanently holding a valid data item

ACM data requirements

- Data coherence
  - Complete data items written or read, not modified mid-transfer
  - Reader does not obtain any reassembled data item that contains parts from different items provided by writer (e.g. you do not get the name of one person and age of another in the same record if ids are being transferred)

ACM data requirements

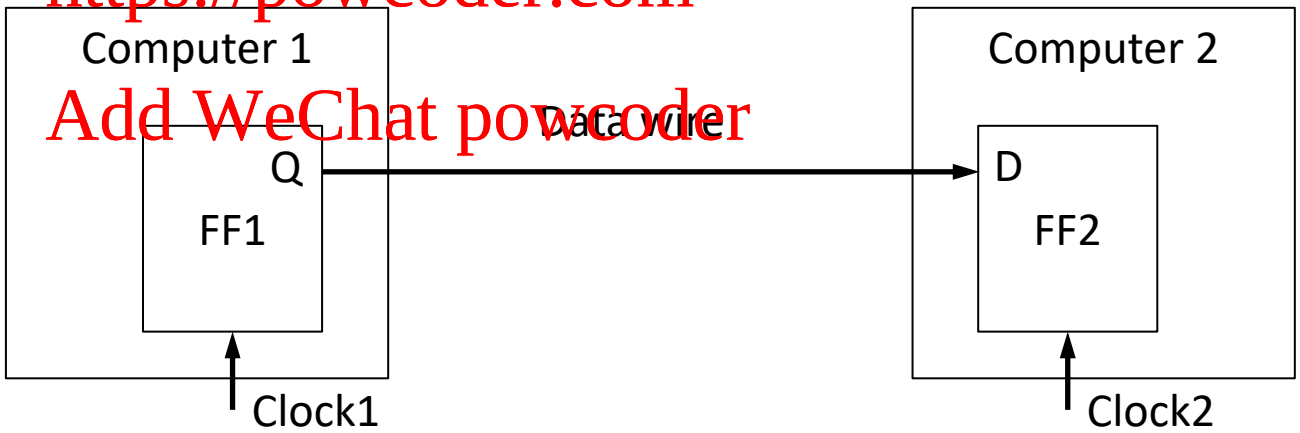
- Data freshness
  - Specific to this pool
  - Reader does not obtain any data item that is older than the most recently fully written item in the ACM
  - Not relevant for multi-stage traditional buffers such as FIFO
- Data sequencing
  - Reader's input data items follow the same order as they are written by the writer (for pool and FIFO)

Data item

- The data communication is assumed to consist of the transfer of a stream of multiple data items
- A data item is a data record/packet/file of the same type/size for each data communication instance
  - The size in particular is not determined without a concrete mechanism and use case
  - Could be quite large – generally assumed not possible to transfer one item during a single reader or writer clock cycle (otherwise the problem is trivial and uninteresting)

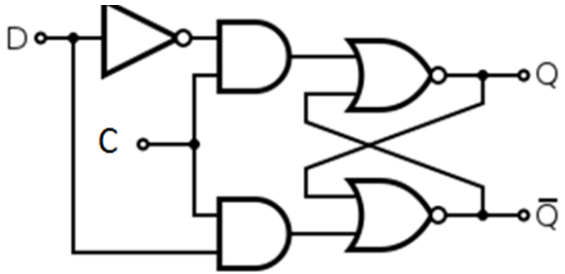
Special case: data item = 1 bit

- To transfer a single bit, a simple method is to copy it from one register to another
  - Writer runs on Computer 1 and reader runs on computer 2



Flip-flop review

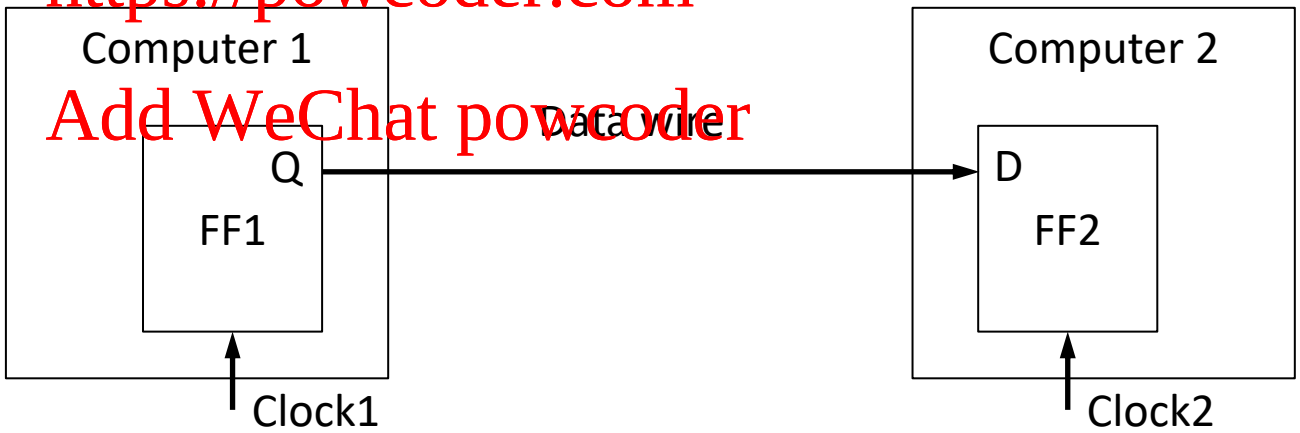
- For a level-triggered flip-flop, when clock=1 (or 0, depending on the design), Q is set to D
- Otherwise, Q keeps its value no matter what D does
- For an edge-triggered flip-flop, Q is only set to D during an active edge of the clock (again the actual active edge could be rising or falling, depending on the design)



D	C	Q_{n+1}	
X	0	Q_n	Hold
0	1	0	Reset
1	1	1	Set

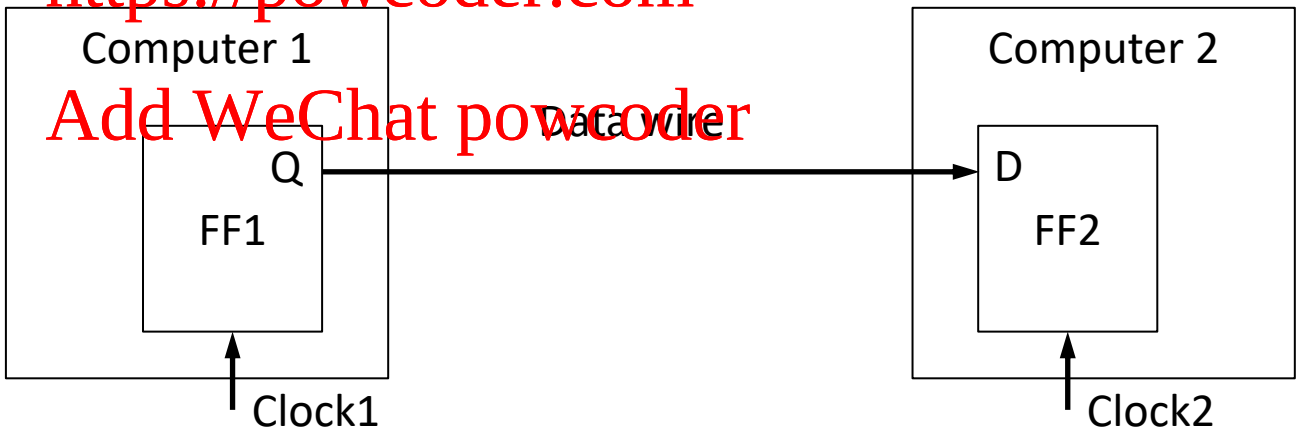
Operations of the FFs

- Writer puts data on wire by setting Q of FF1 using Clock1, reader samples the data on wire using Clock2



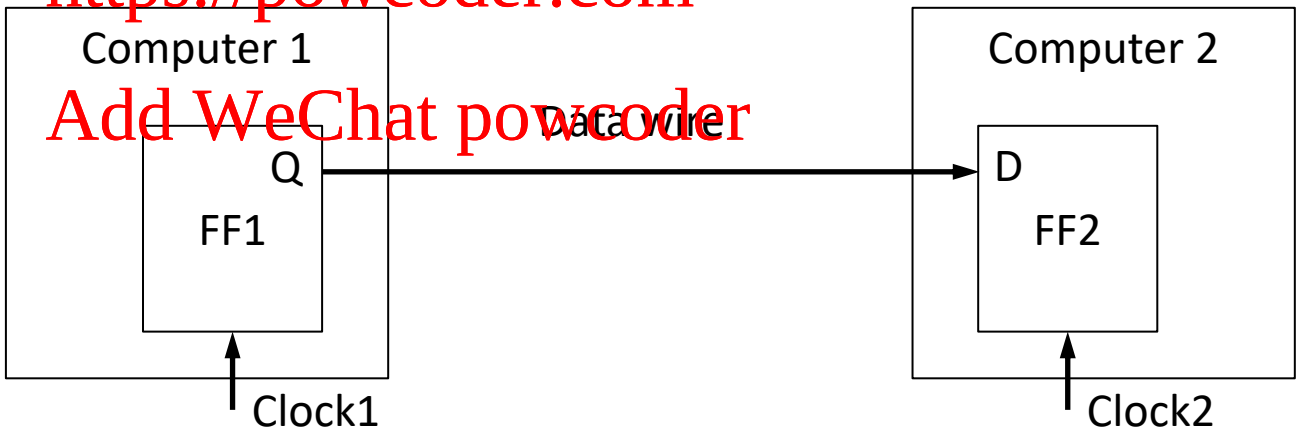
Operations of the FFs

- Assuming rising edge protocol, on the rising edge of Clock2 the value of Q_{FF1} is copied to D_{FF2}



However …

- What if on the rising edge of Clock2 the value of Q_{FF1} is being changed according to Clock1?

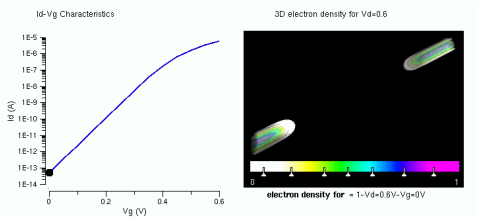


Clocked D-latch requirements

- D-latch is an example, same is true for all types of clocked FFs and latches
  - When you want to sample the data on the D input, relative to the clock edge, that data needs to satisfy "setup and hold" conditions
  - Otherwise you are not guaranteed correct digital behaviour
  - In other words, when you sample the data it must not be in a state of change
  - This is because your reading latch is not a pure digital device, but an analogue device approximating a digital device

Switching element review

- Example: MOSFET
  - A MOSFET cannot switch between 0 and 1 in 0 time
  - Any switch that can switch between 0 and 1 encoded by two discrete values of any physical quantity (voltage, current, displacement, etc.) cannot implement 0-time switching unless it can supply power = ∞, which does not exist
  - Switching in a MOSFET involves the charging or discharging of a non-0 capacitance through a non-0 resistance



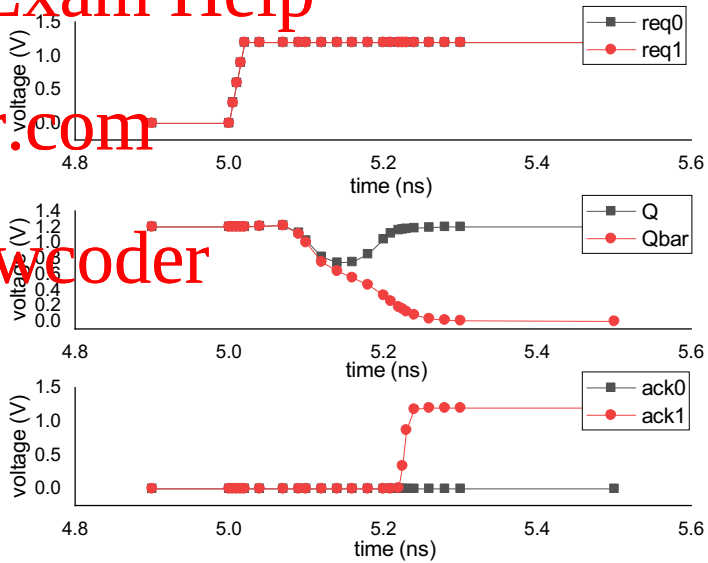
Saumitra R Mehrotra & Gerhard Klimeck

Sampling a wire using a clock edge

- This is an analogue process
  - The clock edge is not vertical in time
  - Clock signal rising/falling takes time ≠ 0
  - The change of signal Q_{FF1} on the writer side is not vertical in time
  - This change takes time ≠ 0
- When these two events are very close or overlap in time
  - The sampled D_{FF2} signal on the reader side is non-deterministic

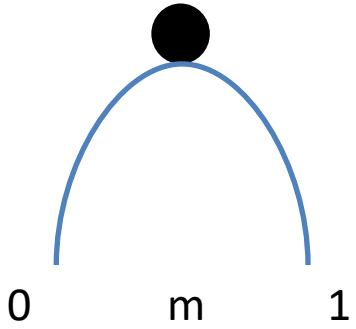
Metastability in synchronizers

- Picture from Prof Delong Shang, IMECAS
  - When sampling clock and signal change clash in time, uncertain D_{FF2} may cause Q_{FF2} to take non-0 and non-1 values for unbounded time
  - FF2 latches this halfway value so that Q_{FF2}=Qbar$_{FF2}$=0.5 (for instance)
  - It is sort of stable but not really, hence the word metastability
  - Q_{FF2} eventually settles to one of the binary values non-deterministically



Mechanical system analogy

- The sampling circuit may be regarded as a wedge/hill entrusted with determining if a ball is falling down the left or right hand half of the space
  - If the ball is on the left hand side, it slides down the left hand side of the hill to a valley on the left, indicating 0
  - If the ball is on the right hand side, it slides down the right hand side of the hill to a valley on the right, indicating 1
  - If the ball is dead centre, however, it may balance on top of the wedge shape for unbounded time, as any wedge, no matter how sharp, is flat/level on top, just as the bottom of the ball it is flat/level



Non-determinism in decision making

- People used to think decision making was deterministic
  - Zhuangzi (C4, BCE) reasoned that he could definitely make a choice between fish and bear's paw by free will
  - Aristotle (C4, BCE) ridiculed the idea that if a man is equally thirsty and hungry, and given food and drink, he could die of hunger/thirst because of indecision
  - Al-Gazhali (C11-12, CE) thought that a man possesses an inherent quality the nature of which is to differentiate things, therefore two seemingly the same dates with equal desirability will not impede the choosing of one

Non-determinism in decision making

- Jean Buridan, c. 1340
  - 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.
  - The (in)famous "Buridan's ass" paradox
  - Picture credited to New York Herold, c. 1900

What about the maths?

- Metastability has exponential decay
  - In other words, the longer the time, the more likely any metastability has settled, but the settling probability increase is highest in the beginning (near time = 0) but reduces towards time = ∞
- Mean time between failure of a synchronizer because of metastability

$$MTBF = \frac{e^{t_r/\tau}}{F_D \cdot F_C \cdot T_p}$$

T_p propagation delay of FF2
F_D data frequency (FF1)
F_C clock frequency (FF2)
t_r resolution time (sync period)
τ time related to setup and hold (FF2)

In other words

- MTBF is greater (better) if
  - FF2 is faster (sharper wedge/hill in the mechanical analogy)
  - Both data change and sampling clocks are slow
  - Resolution time requirement is relaxed (you can wait for a longer time before synchronization must be completed – usually realized with multiple layers of FFs on the reader side)
- In other words, you need to use fast semiconductor slowly
  - When you read claims that someone's metastability MTBF is longer than your lifetime or even the lifetime of the universe, be very suspicious

In other words

- Asynchronous data transfer does not always work even for a single bit
  - How can we then organize large-size data items to be passed from writer to reader?
- Turns out this is overly pessimistic ☺
- Question:
  - Why do we seem to only care for the settling of the signal/data (potentially unbounded delay) and not for the value it settles to (non-deterministic 0 or 1)?