Asynchronous or Synchronous?
That is not the question!

Scott Fairbanks and Simon Moor
**DATA CONTROL SPACE**

**ASYNCHRONOUS**
- Locally derived timing signals
- Non deterministic scheduling
  - Tokens
- Timing signal bundled to data /variable periods
  - latches
  - bit or word per timing signal

**SYNCHRONOUS**
- Timing signals derived from a single source
  - deterministic scheduling
    - Periods
- Data signal bundled to clock /variable periods
  - flip flops
  - global or phased timing signal
DATA CONTROL SPACE

DISTRIBUTION
- Generation or Amplification

SCHEDULING
- Deterministic or Non deterministic

INITIATION
- Periods
- Tokens
- Data tokens
- Period tokens

STATE HOLDING
- Latches, Domino or Flip flops

PRECEDENT
- Data signal or Timing signal

SCOPE
- Bit, word, phase, global/regional
Must be some balance of amplifying timing signals or locally generating timing signals
SCHEDULING

Deterministic
or
Non deterministic

Flexibility and arbitration?

Scheduling, certainty and rigidity?
INITIATION
	Tokens
	Data tokens
	Period tokens
	Persistent, reactive, simultaneous?

Does token mean new data or potential for new data?

Are the tokens persistent?
!traditional clocked and de synchronized

Simultaneous !traditional clocked
Philosophical question: What bends, timing signal or data delays?

Clocked, GasP timing signal precedence
Bundled data, DI data delay precedence
SCOPE
Bit, word, phase, global/regional

Bit
DI/QDI

Word
Micropipelines
GasP

Phase
Domino
Skew tolerant domino

Global
Clock
Flip Flops  simplest, most robust designs but poorest performance.

Latches  Better occupancy, more complex timing margins.

Domino  Fastest, most complex timing and has electrical issues.
## EXAMPLES

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>GasP</th>
<th>QDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling</td>
<td>Deterministic</td>
<td>Non deterministic</td>
<td>Non deterministic</td>
</tr>
<tr>
<td>Initiation</td>
<td>simultaneous, period tokens</td>
<td>data tokens</td>
<td>data tokens</td>
</tr>
<tr>
<td>Precedent</td>
<td>timing signal</td>
<td>timing signal</td>
<td>data signal</td>
</tr>
<tr>
<td>Scope</td>
<td>global</td>
<td>word</td>
<td>bit</td>
</tr>
<tr>
<td>State Holding</td>
<td>flip flops mostly</td>
<td>latches</td>
<td>latches</td>
</tr>
<tr>
<td>Distribution</td>
<td>amplification</td>
<td>mostly generation</td>
<td>generation</td>
</tr>
</tbody>
</table>
Unexplored permutations?
CONCLUSION

- I argued that current categories for data control systems limit creativity, are inaccurate, and are misleading.

- I introduced six mostly independent variables for data control systems.

- Provided framework for systematically exploring data control space.