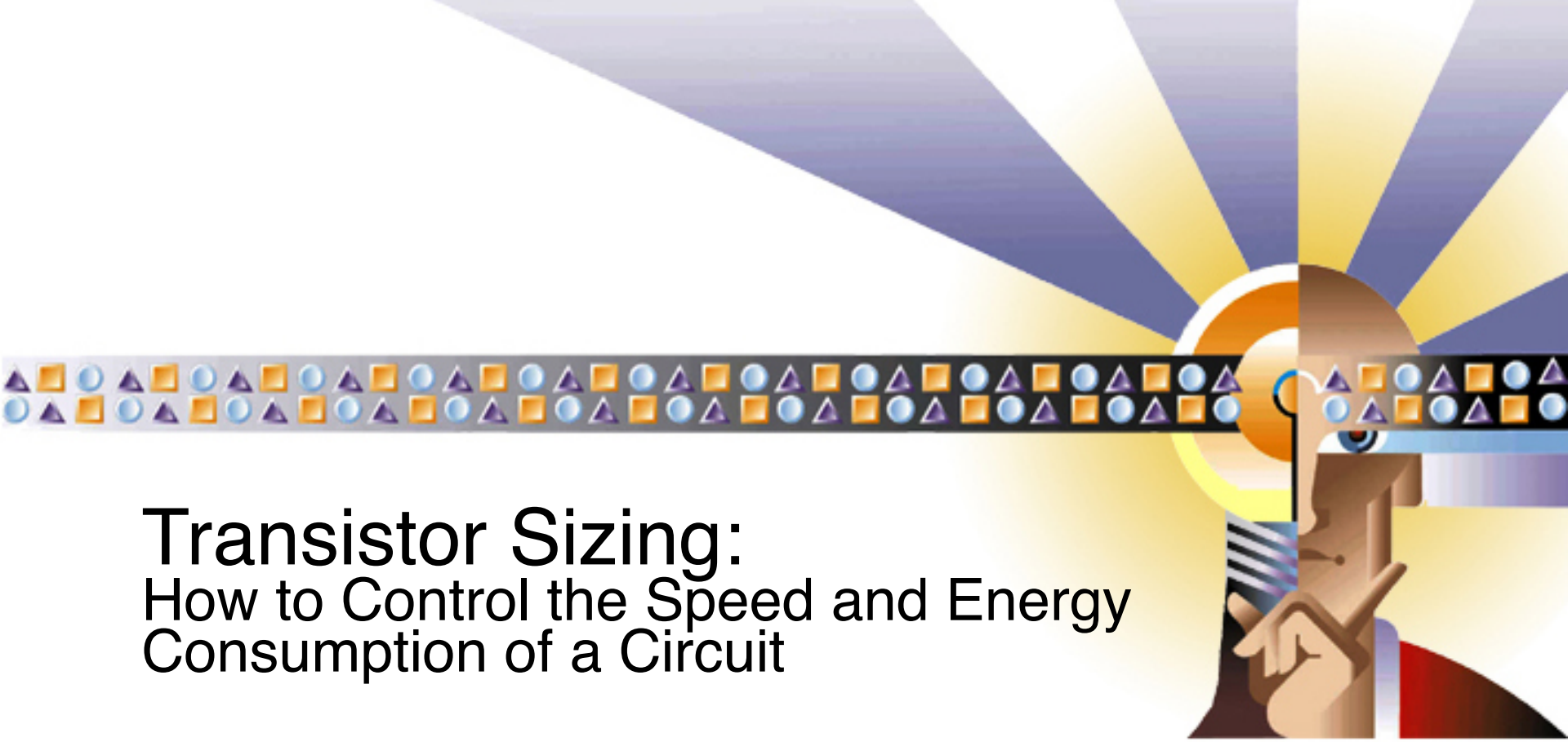


Transistor Sizing:
How to Control the Speed and Energy Consumption of a Circuit

Jo Ebergen, Jonathan Gainsley, Paul Cunningham

Async Design Group

Sun Labs

SML2004-0325

Transistor Sizing: How to Control the Speed and Energy Consumption of a Circuit

Jo Ebergen, Jonathan Gainsley, Paul Cunningham
Asynchronous Design Group
Sun Labs

Introduction

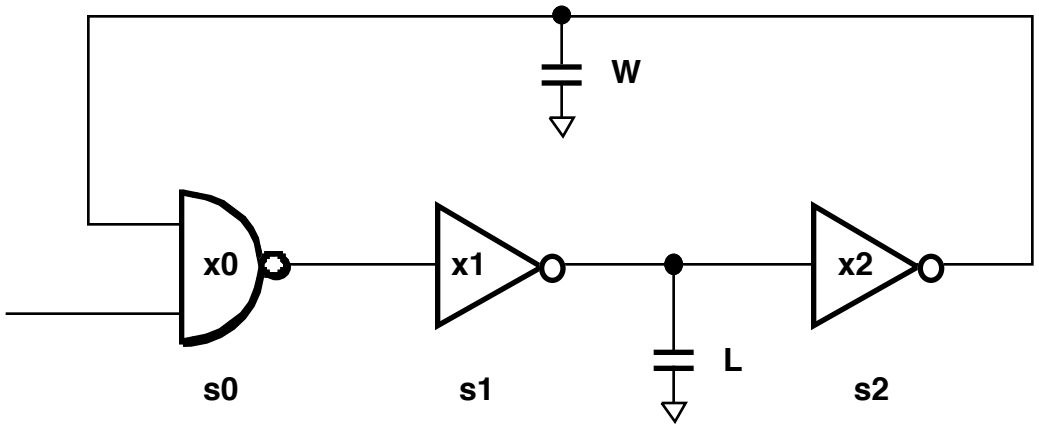
- Transistor sizes (widths) determine
  - Speed of circuit
  - Energy consumption
  - Total area of circuit
  - Satisfaction of delay constraints
- Success or failure

How Do I Pick Transistor Widths?

- To optimize for speed?
- To optimize for energy?
- Automatically and quickly?
- Does a circuit have a speed limit?
- Is there a trade-off between speed and energy?
- How do I compare circuits built for different technologies?

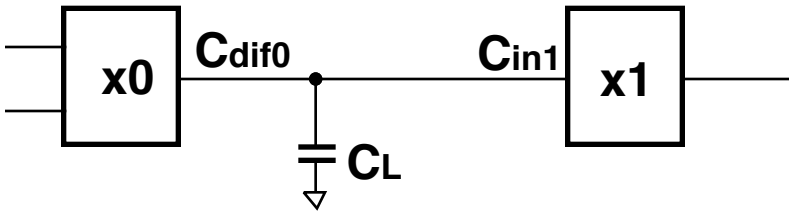
An Example

- Given desired gate delays s0, s1, and s2, fixed latch load L and fixed wire load W
  - How do I find the sizes x0, x1, and x2?
- Cycle time = s0+s1+s2. What is minimum?



The Delay Model

- Defines relationship between gate sizes and delays
- Capacitance driven by gate in time s
  = sum of all capacitances on node
  - s = gate delay
  - x = drive strength [capacitance/time]



$$s0*x0 = C_{dif0} + C_L + C_{in1}$$

- Input and diffusion capacitances are linear functions of drive strength [Idea of Logical Effort]

Units

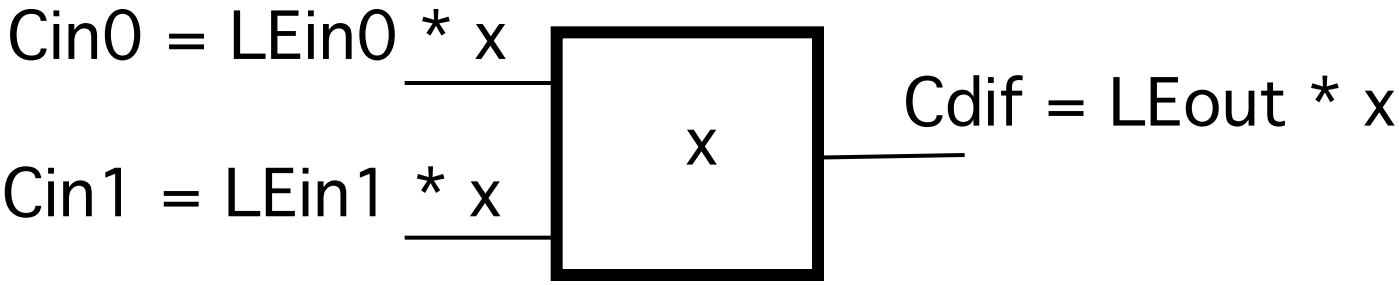
- Unit of capacitance
  - κ = Input cap of min. sized inverter
  - All fixed loads must be converted
- Unit of delay s (for stepup or slope)
  - τ = Delay of ideal inverter, with no diffusion capacitance, driving copy of itself.
  - FO4 = 5*τ
- Unit of drive strength x (as in 4X, 8X)
  - κ/τ = Capacitance per time unit

Technology Independence

- Units κ and τ depend on technology
- Ex: TSMC 180nm, τ =17ps (FO4 =85ps)
- Equations are independent of technology
- Allows comparisons of circuits in different technologies
- Warning: wire loads do not scale linearly

Logical Efforts

- Let me find gate capacitances as function of size x
- Input and diffusion capacitances are proportional to drive strength
- Logical Effort of input (LEin) =
  input capacitance per unit of drive strength
- Logical Effort of output (LEout) =
  diffusion capacitance per unit of drive strength

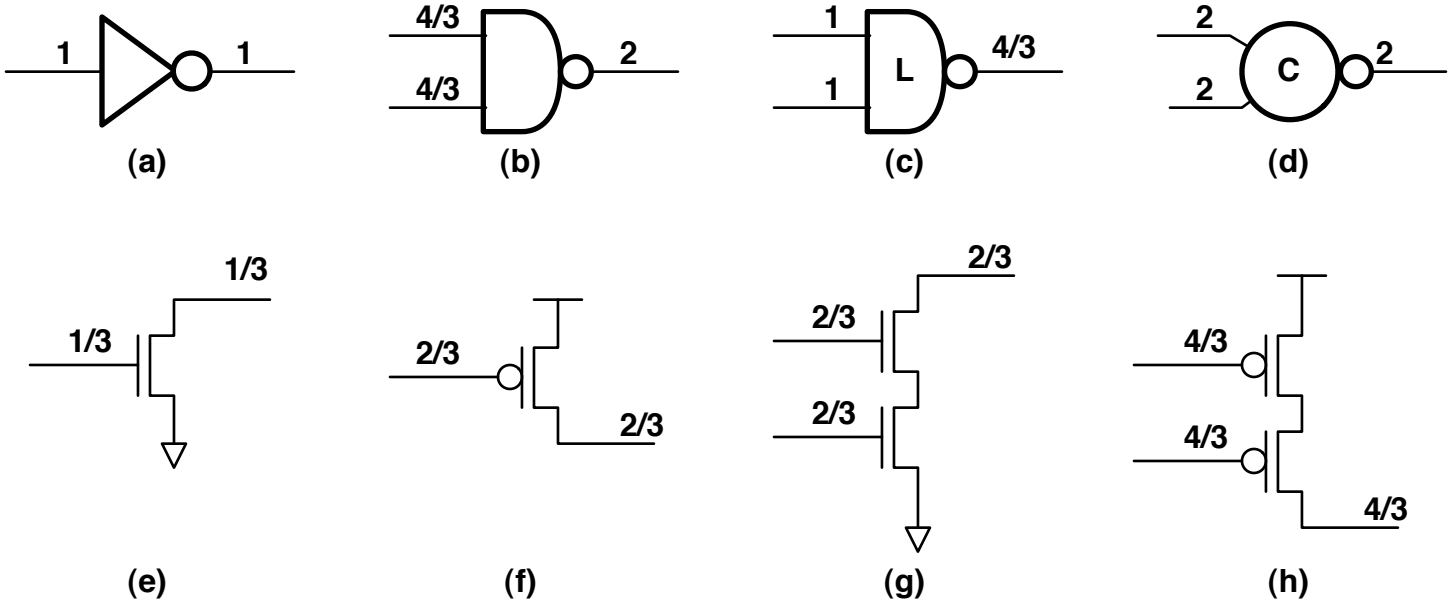


Properties of Logical Effort

- Logical Effort =“effort” to compute logic function
- Logical effort is a time constant [$\kappa/(\kappa/\tau)=\tau$]
- LEout = time to load diffusion capacitance of gate
  - parasitic delay
- LEin = time to load input capacitance of gate (with the same drive strength)
- Logical efforts can be found from transistor diagram or empirically

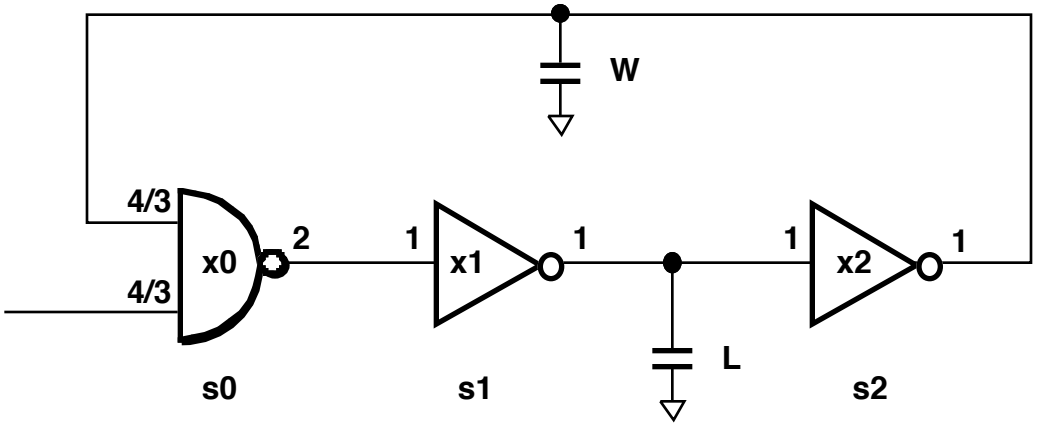
Examples

- Some gates and their logical efforts



Back To Example

- Use delay model, now with Logical Efforts



$$
\begin{array}{l|l|l|l|l}
s0 * & x0 & = & 2*x0 + 1*x1 & \\
s1 * & x1 & = & 1*x1 + 1*x2 & + L \\
s2 * & x2 & = & 4/3*x0 + 1*x2 & + W
\end{array}
$$

In Matrix form: $S * \underline{x} = T*\underline{x} + \underline{C}$

In General

- S*x = T*x + C
- S = diagonal matrix of gate delays
- x = vector of drive strengths
- T = logical effort matrix
  - Tij=0 no connection from gate i to gate j
  - Tij≠0 connection from gate i to gate j
  - Describes topology of circuit
- C = vector of fixed loads
- Equations can be extracted from netlist

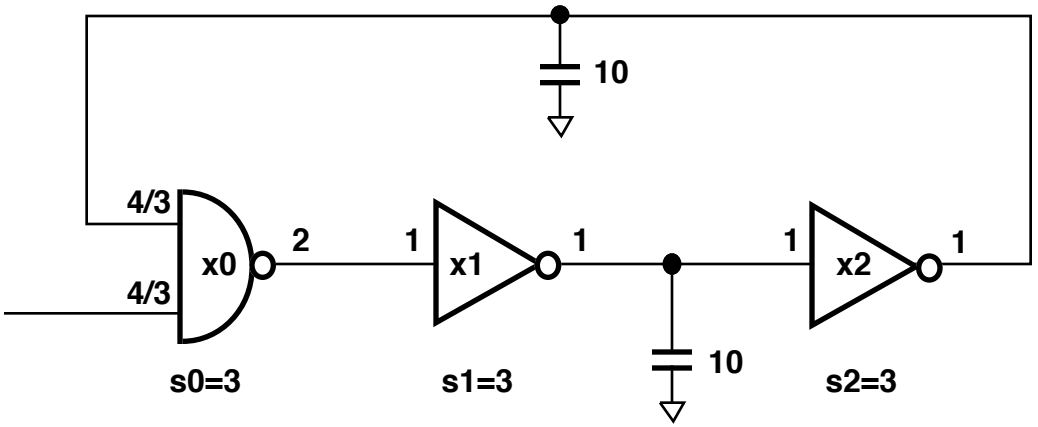
Now What?

- Does a solution x exist for given S, T, and C?
- How do I compute solution for x efficiently?
- How to choose delays S?
- Special case: equal gate delays
  - Simpler model: s*x=T*x+C
  - Path delay = (# gates)*s
  - Easy for satisfying delay constraints
  - More accurate

How To Compute Drive Strengths?

- Many ways to solve s*x=T*x+C
- Easiest is iteration
- Let f(x)=(T*x+C)/s
- x(0)=0
- Repeat x(i+1)=f(x(i)) until convergence
- Converges quickly, if you choose s well

An Example



x2=(x2+10+4/3*x0)/3
x1=(x1+10+x2)/3
x0=(2*x0+x1)/3

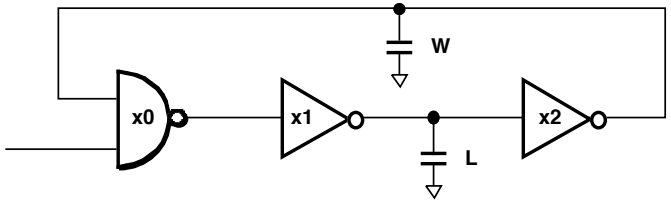
x2:	0	3.66	4.69	4.74	..
x1:	0	4.89	5.23	5.25	..
x0:	0	2.3	2.41	2.41	..

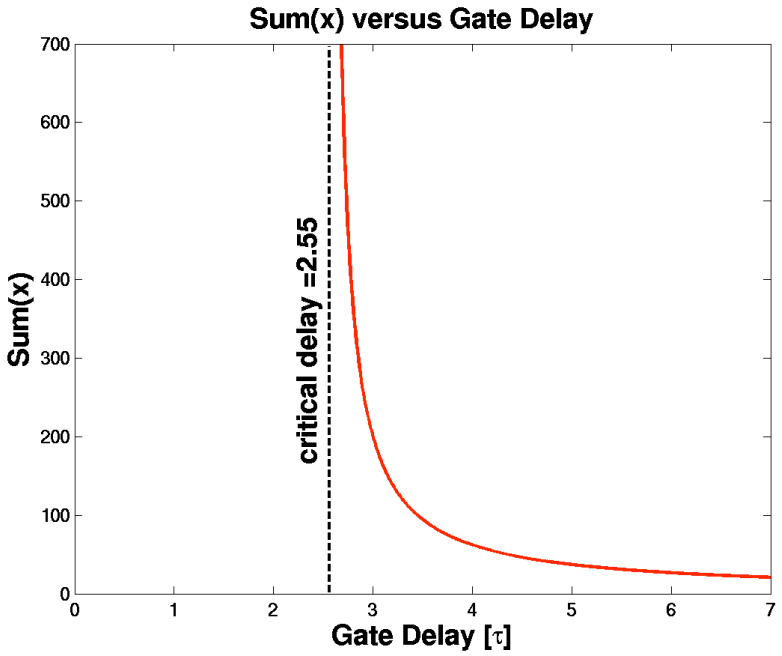
Critical Delay

- Only for circuits with cycles
  - Almost all async control ckts have cycles!
- Equal gate delay: $s*x = T*x + C$
- Critical delay (cs) of circuit
  = largest real eigenvalue of T
- Feasible solution exists iff s is larger than critical delay ($s > cs$)
- Critical delay is independent of fixed loads C
- Sizing algorithm converges if $s > cs$

Total size and critical delay

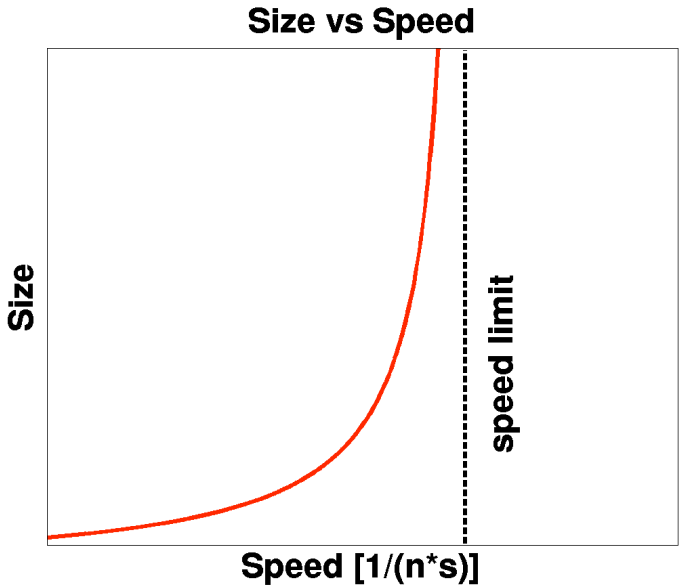
- Total size ($\sum$x's) as function of gate delay
- Size grows as C/(s-cs)





Critical Delay and Limits

- Critical delay defines lower bound for gate delay, assuming equal gate delays
- Critical delay (cs) and # gate delays (n) in cycle define speed limit for throughput =1/(n*cs)



Energy Estimation

- Dynamic Energy
  - Due to (dis)charging capacitance C
  - Proportional to $C*V^2$
- Short-Circuit Energy
  - Due to crossover current
  - If input and output slope are equal, short-circuit energy ≈ α * dynamic energy consumption [Veendrick84]
- Static Energy
  - Due to leakage currents
  - Ignore for now

Units

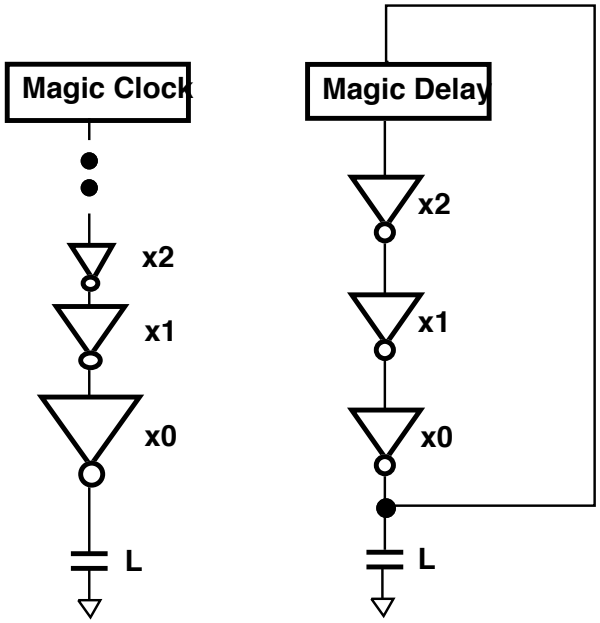
- Unit of energy
  - ε = energy spent by ideal minimum inverter, excl. diffusion capacitance, driving a minimum inverter
- "Energy spent" = energy lost in resistors
- Unit ε depends on technology, but equations do not
- Can be determined empirically
  - ε=2.9fJ in TSMC 180nm, 1.8V.

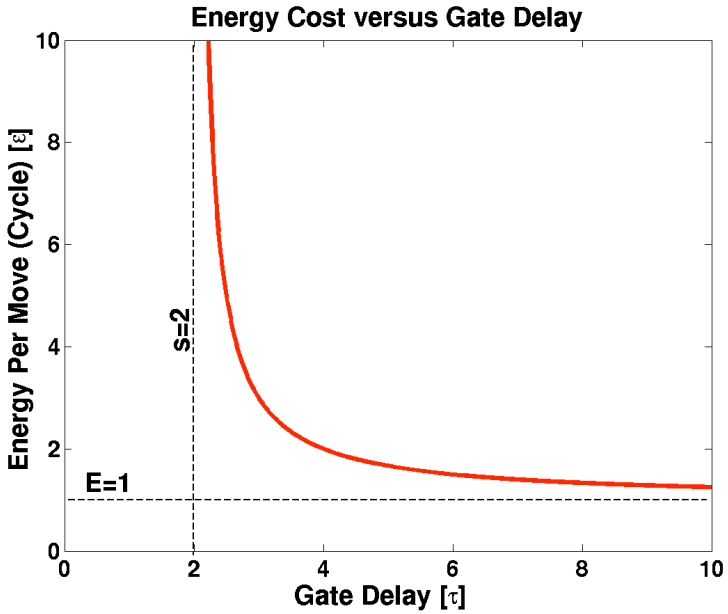
More On Energy Estimation

- In equal-gate-delay model
- $s^*x=T^*x+C$
- Energy spent by gate $\propto$ total output cap
- Energy spent by gate i $\propto (T^*x+C)_i = s^*x_i$
- Let p_i be activity index of gate i in an execution
- Total energy spent in an execution = $\sum_i p_i^*s^*x_i$

An Example: Magic Clock

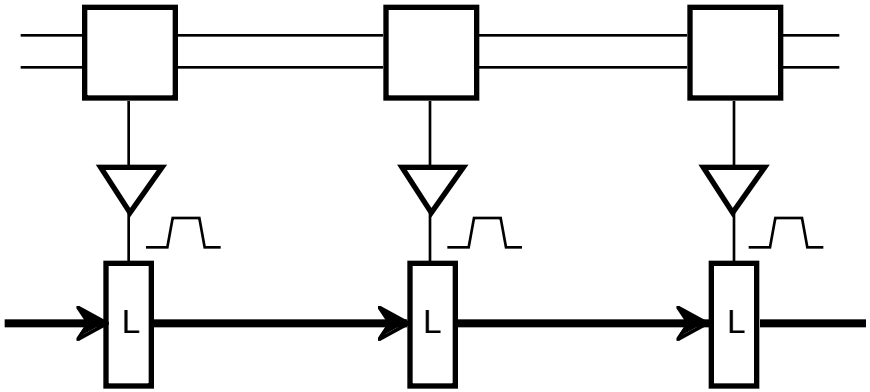
- Every inverter has the same delay
- Control must charge and discharge load L
- E = 1 + 2/(s-2) per unit load
- For equal delays, the best you can do





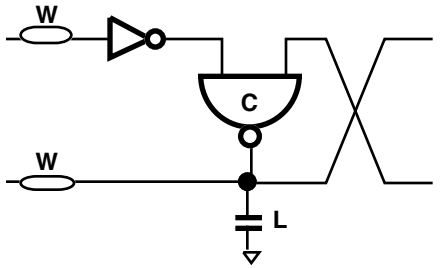
Comparing Circuits

- Independent of process technology
- Example: asynchronous controls of ripple FIFO
- How do different implementations compare in terms of energy versus performance?

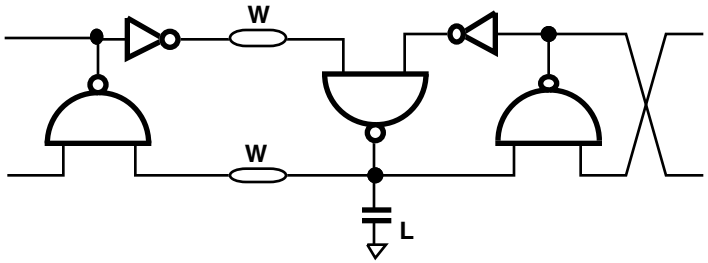


The Implementations

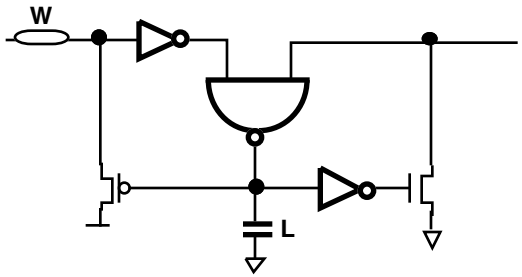
- Chain of Rendezvous (COR):



- asP*:



- GasP:



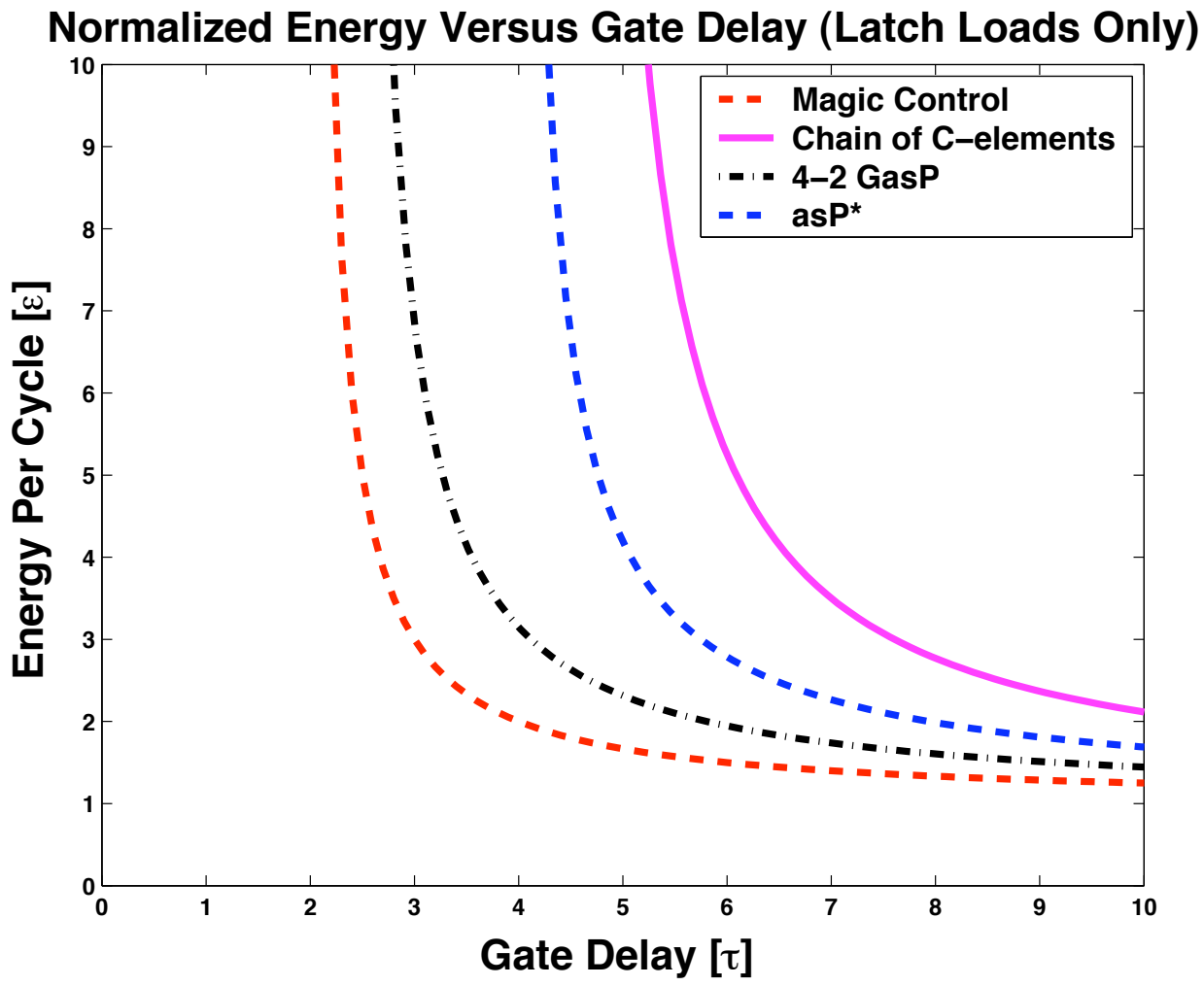
More Implementations

- Berkel's single-track handshake
- Singh & Nowick's High-Capacity Pipeline
- IPCMOS by Schuster et al
- ....
- Magic clock
  - the lower bound and the ideal "synchronous" implementation.

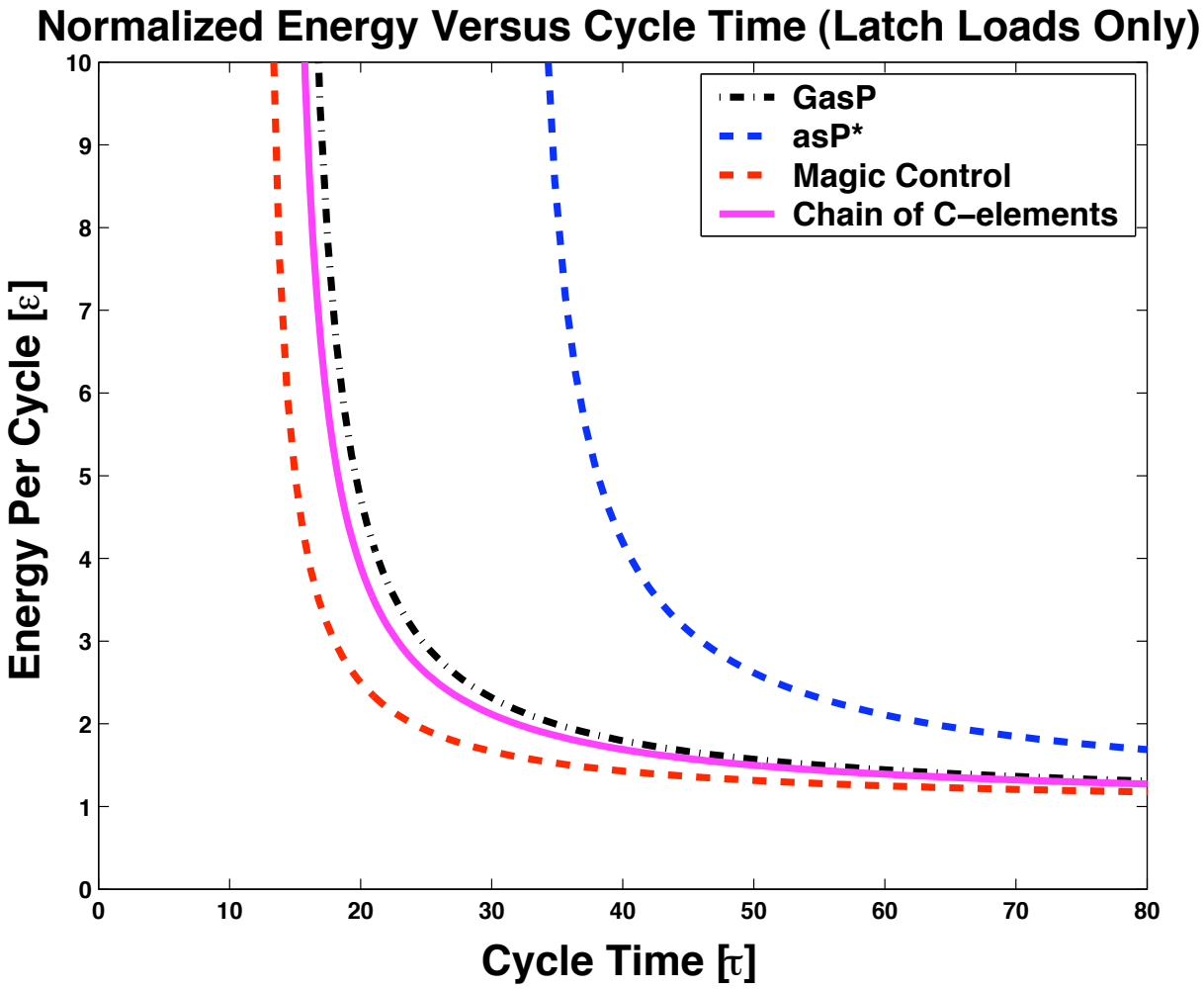
Critical Times

Ckt	Critical Gate Delay	# Gates in Cycle	Critical Cycle Time
Magic Clock	2.0	6 [1)]	12
GasP	2.42	6	14.52
IBM IPCMOS [2)]	2.51	14	35.14
Singh-Nowick	3.38	8	27.04
asP*	3.95	8	31.6
Berkel's single-track	4.21	6	25.26
Chain of C-elements	4.56	3 [3)]	13.68

A Price/Performance Comparison



A Price/Performance Comparison



It's (Like) The Economy, Stupid

- Moving charges in circuit
  = Moving capital goods in economy
- Open input-output model
  - "Gate" = economic sector
  - "Capacitance" = demand for capital goods
  - "Drive strength" = supply of capital goods per time unit
  - "Energy" = total cost of capital goods
- Wassily Leontief (1906-1999)
- Has many applications
- Abundant literature
- A chip is like an economy

Summary

- Simple model for calculating transistor sizes
- Simple and efficient algorithms
- Good results obtained so far with equal gate delays
- Critical delay gives a price/performance characteristic for a control circuit
- Gives insight into speed-energy trade-offs
- Allows comparisons of circuits independent of process technology