Exhibit F
(8 of 12)

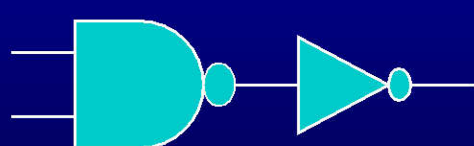
That's very nice in theory, but….

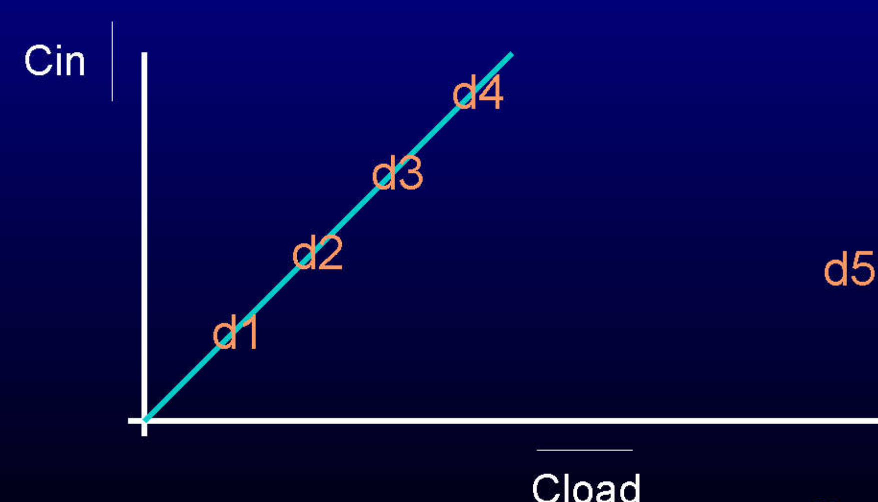
- Library only has a few drive strengths: is there a descretization error?
- How to account for differences in fall and rise time?
- Do I need a special library?
- What if a very large drive strength is needed?
- When are buffers inserted?
- Isn't the model too simplistic?
- What about the parasitic wire resistance?

Library Analysis

/cmos18/NAND2 (A -> Z) inverting

model	hide	typ load	gain	input cap	area	rise delay	fall delay	slew	max slew
NAND2d1		25	2.51	10	1	161	102	66	2000
NAND2d2		54	2.71	20	1	153	100	67	2000
NAND2d3		110	2.69	41	2	153	100	67	2000
NAND2d4		186	2.66	70	5	153	99	67	2000
NAND2d5	D	370	18.52	20	9	254	293	57	2000
NAND2_SUPER		370	2.74			148	108	67	2000

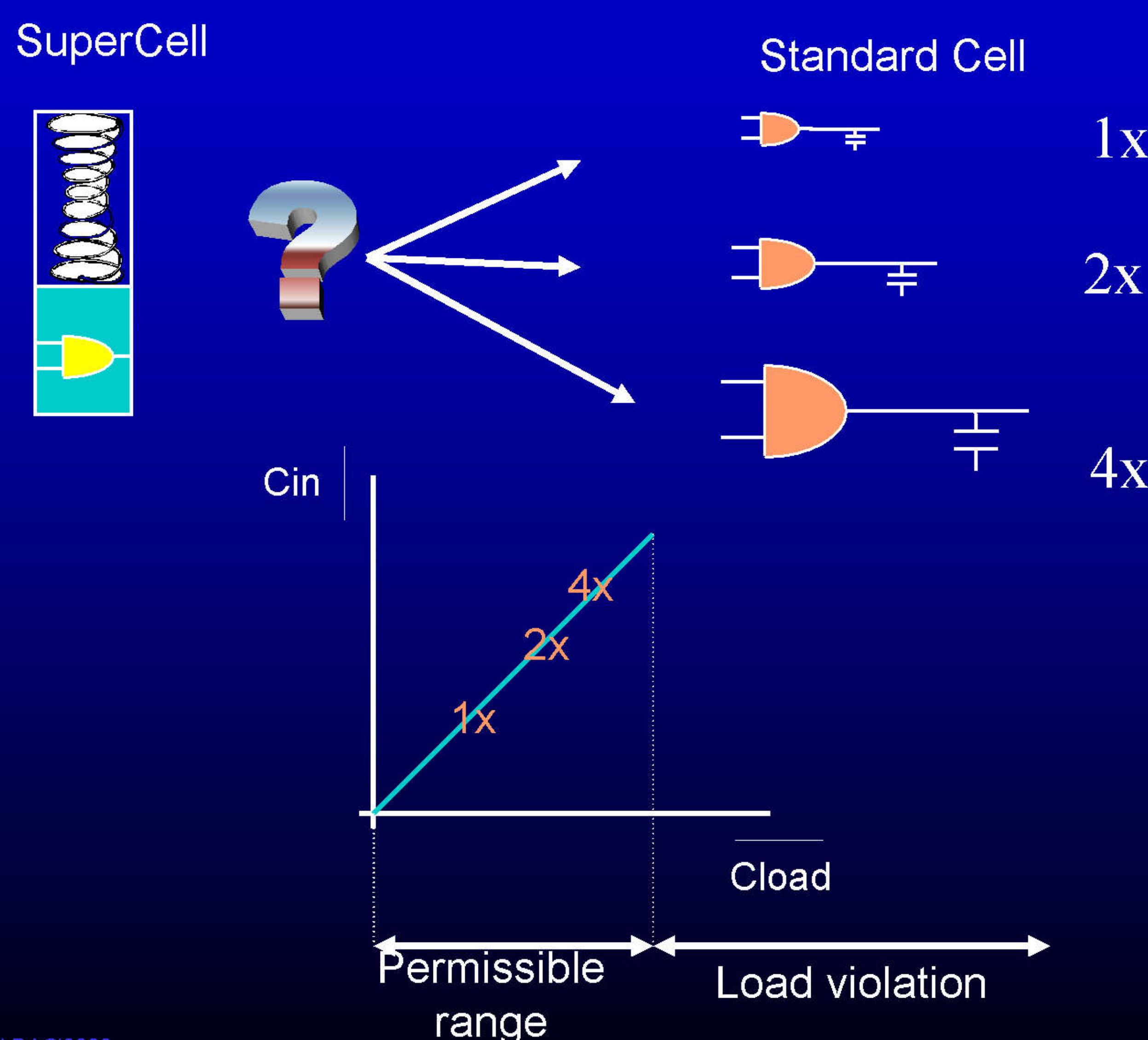




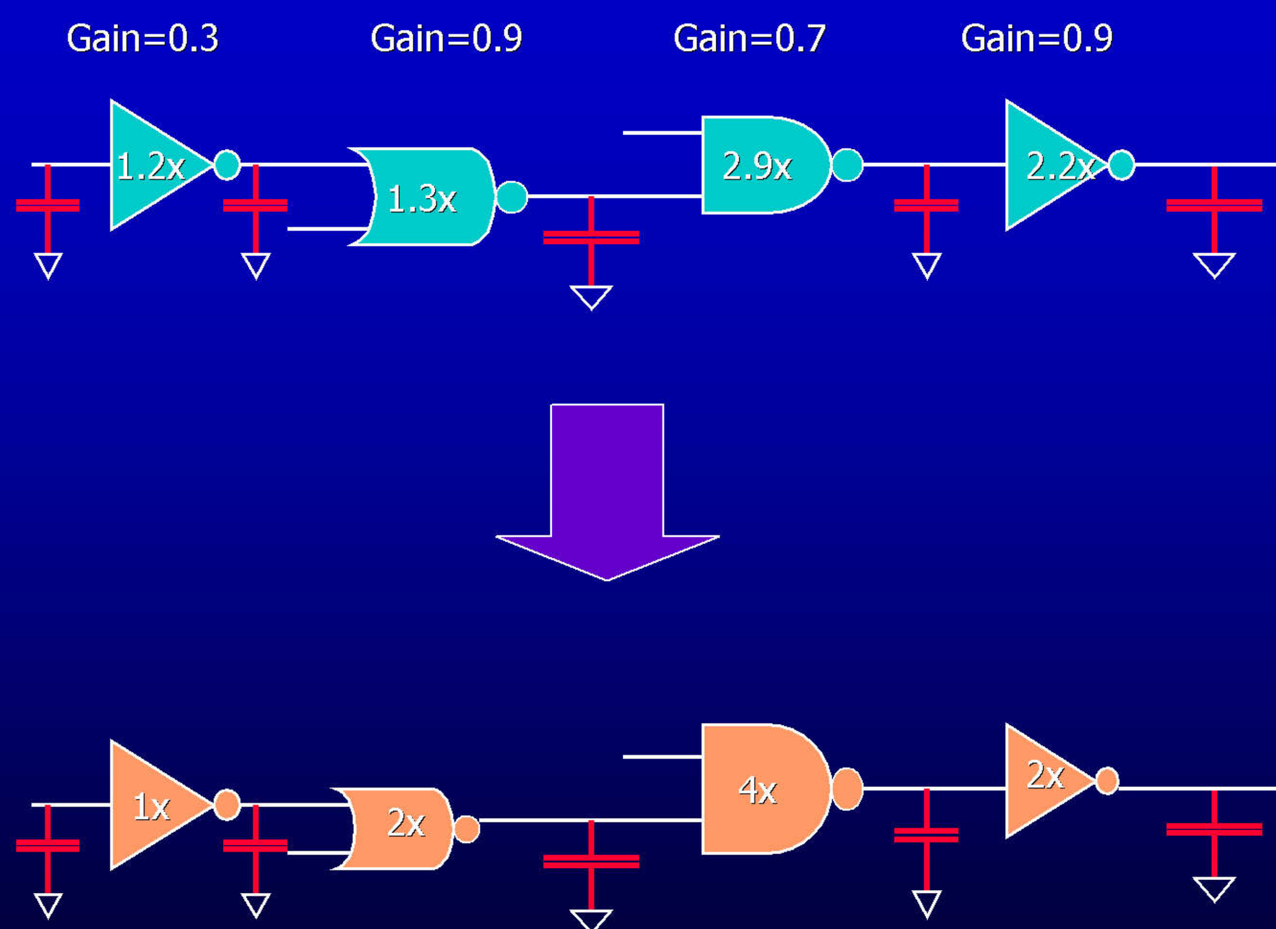
- Gain is averaged
- Toss out 'weird cells'
- Typical load is the load the gate drives when optimized for maximum speed: g*h =3.59

MAGMA0791089

Fixing cell sizes & keeping timing

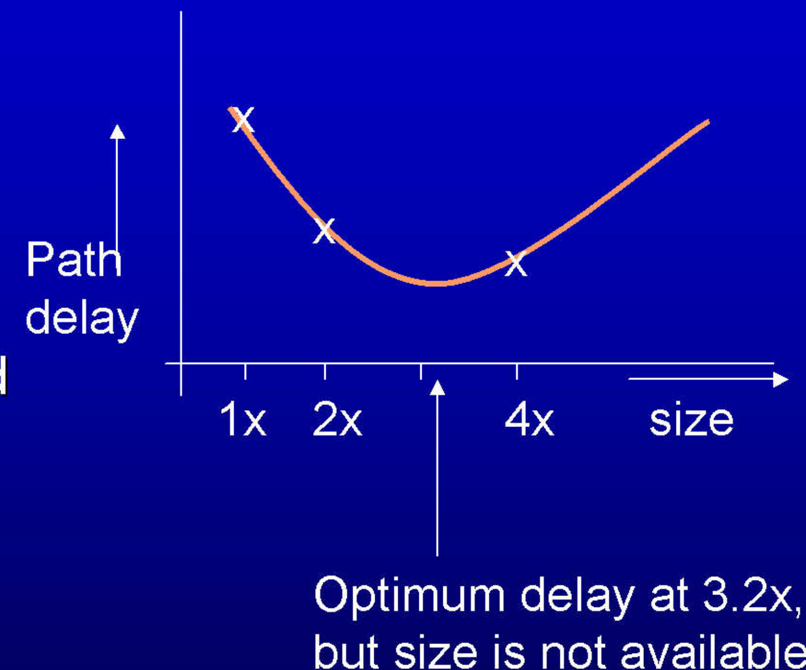


The discretization error...

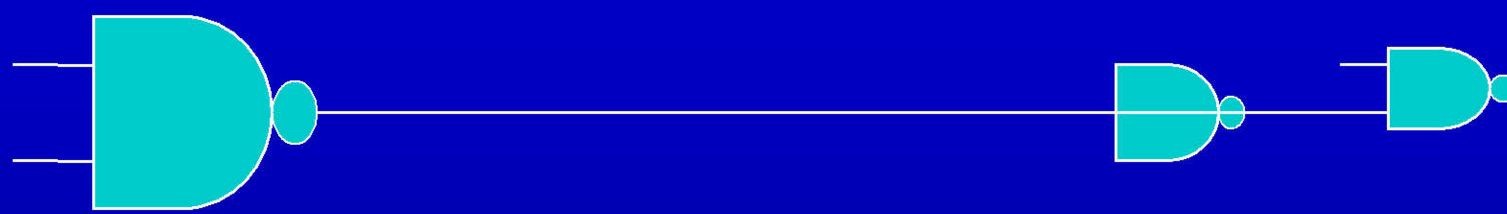


.. is generally not a big problem

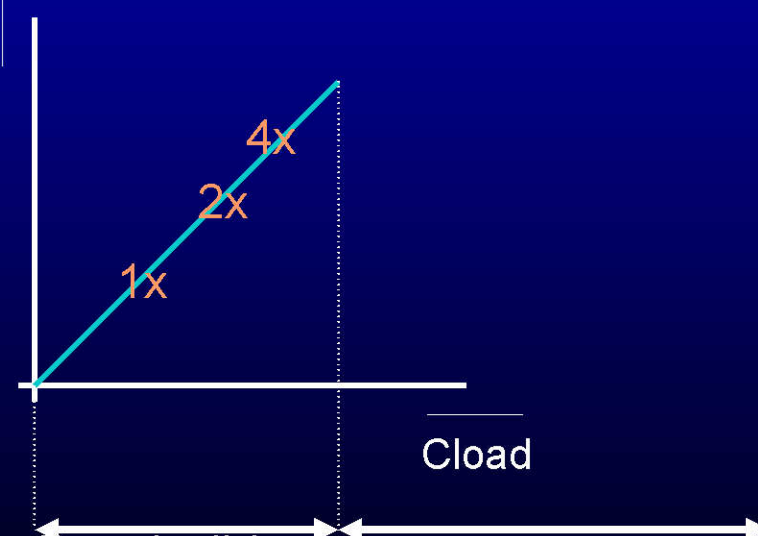
- Delay versus size curve is flat, because the size is optimized for maximum speed
- Rounding error is absorbed by appropriate up- and downsizing of surrounding cells.
- On critical paths, buffer insertion and logic restructuring minimize effect.



Load violations



- Maximum drive strength in the library might be too small
- Drive information is stored in super cell, and managed pre-placement.
- Buffering, cloning and restructuring are used to maintain delay during placement



Timing closure

Buffered wire: smallest delay

- Delay per stage (elmore):

$$d = \frac{R_0}{w}(C_w L + C_0 w) + \frac{R_w C_w L^2}{2} + R_w L C_0 w$$

- Optimum buffer distance:

$$L_{opt} = \sqrt{\frac{2\tau(1 + p_{buffer})}{R_w C_w}}$$

- Optimum buffer size:

$$w_{opt} = \sqrt{\frac{\tau C_w}{R_w C_0^{\,2}}}$$

Buffering in a typical 0.25 μm process

- Optimum buffer distance tends to be around 2000 μm.
- This works out to an area of 4mm^2, or about 10-20K cells.
- But w_{opt} is *much* larger then what most libraries have available: