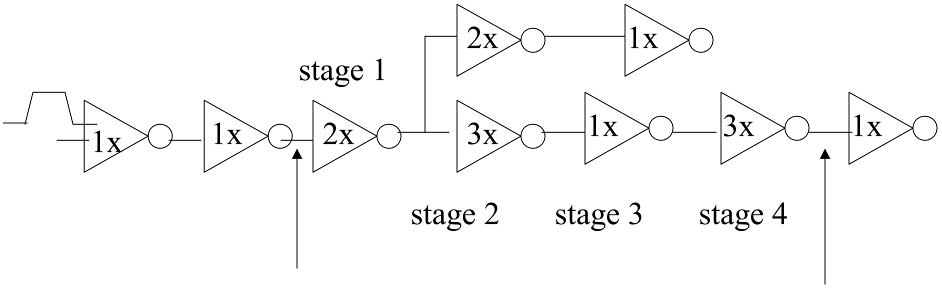
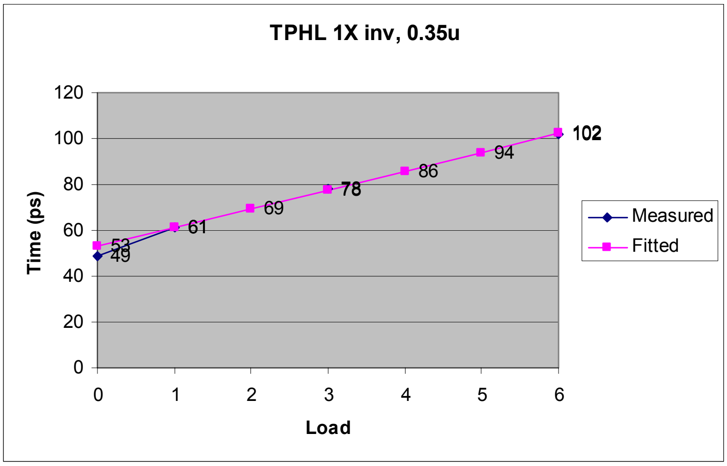
Goal: Predict this delay



Tplh (path) = Tphl(s1) + Tplh(s2) + Tphl(s3) + Tplh (s4)

Tphl (path) = Tplh(s1) + Tphl(s2) + Tplh(s3) + Tphl (s4)

Part #1 Data



Typical Data, line fitting: delay = 8.2 * Load + 53

K fitted values, measured noload values

Driving Gate Size	tphl		tplh	
	K	noload	K	noload
1X	8.2	53	12.6	57
2X	6.3	61	7	63
3X	5.2	72	5.3	68
6X	3.2	101	3	89

delay = K * Load + noload

Load is measured in inverter gate sizes.

Question: How do you compute load?

Because we are only using one type of gate, can just use sum of inverter gate sizes as load value.

Load of stage1 is 5; K values decreasing with driving gate size.

Can also compute Load as
output gate sizes/driving gate size

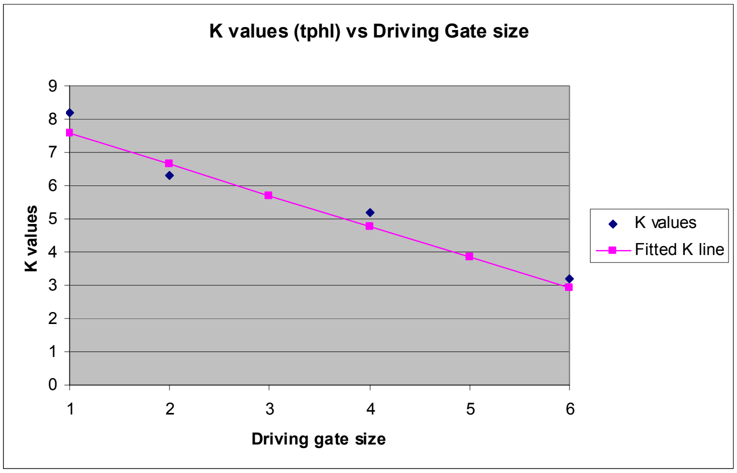
In this case, Load of stage1 is 5/2 = 2.5; K values will be increasing with driving gate size.

BOTH methods calculate the SAME DELAY as long as you are consistent.

Part #3, Use a single K value

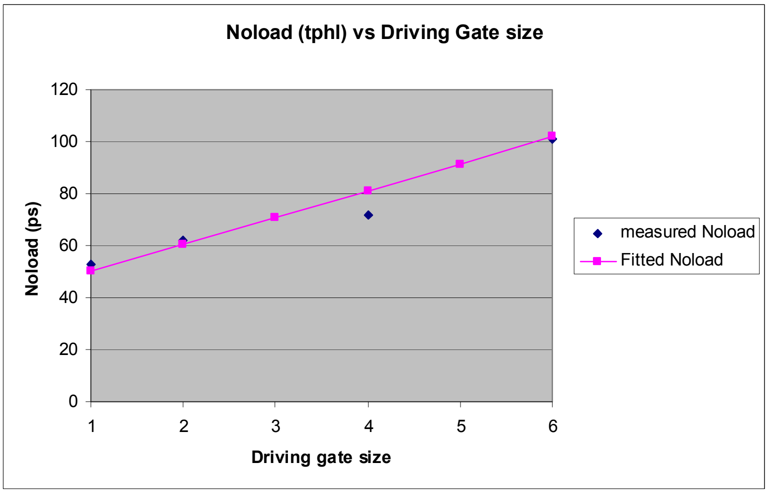
- How do we pick two K values and two noload values? – one for tplh and one for tphl??
- Approach #1: Pick a “typical” gate and use that
  - Just use K, noload values for 3X gate
- Approach #2: Relate K values to driving gate size so that we can compute ‘K’ based on driving gates
  - A common method is to let K of Gate Nx = N * K of gate 1x
  - Another method is to fit K’s, noloads to straight line or some other curve
    - Instead of remember K, noload values, remember equation parameters to compute K, noload based on size.

Fitting K values to driving gate size



K value = m * gate_size + b = -0.9 * gate_size + 8.5

Fitting noload values to Driving Gate size



noload = m * gate_size + b = 10.3 * gate_size + 40

Delay Predictions

	TPHL	TPLH
Single K, noload (K for gate size = 3)	-1.1%	7.2%
Computed K, noload based on driving gate size	2.8%	-3.2%
Actual K, noload	3.0%	1.0%

Looks pretty good..... but what about other paths?

Try 3 new paths: all gates on path = 1x; 3x; 6x

Delay Prediction - 1X gates on path

	TPHL	TPLH
Single K, noload (K for gate size = 3)	-0.7%	3.5%
Computed K, noload based on driving gate size	-14.1%	-14.3%
Actual K, noload	-4.8%	-10.3%

Under estimating the delay

Delay Prediction - 3X gates on path

	TPHL	TPLH
Single K, noload (K for gate size = 3)	23.3%	17.5%
Computed K, noload based on driving gate size	29.1%	15.8%
Actual K, noload	23.3%	17.5%

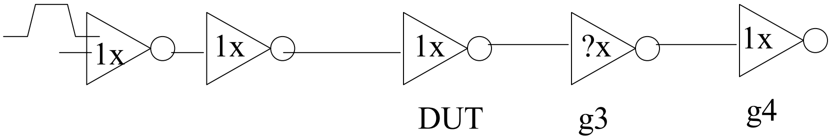
Not good – over estimating delay significantly

Delay Prediction - 6X gates on path

	TPHL	TPLH
Single K, noload (K for gate size = 3)	38.6%	27.9%
Computed K, noload based on driving gate size	48.1%	28.6%
Actual K, noload	47.4%	41.5%

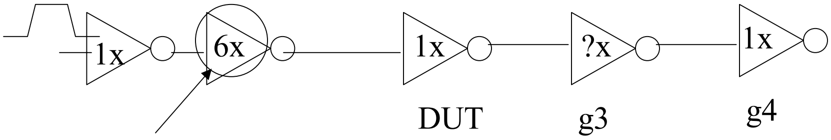
Delay estimations off by almost 50%!!!

Why such a large error??



Original data for K's taken with above test circuit.

Will get significantly different Ks if test circuit below is used:



Note the difference!! The DUT will experience significantly different input slopes between the two test circuits.

So, what to do?

Most modern synthesis tools use a lookup table approach for delay estimation where the two axis of the table is input slope and output load. Must have this table for EACH GATE SIZE, and for EACH DELAY (different tables for TPLH, TPHL

Delay table for 1X inverter, TPLH

Input slope (V/ps)	Output Capacitance Load			
	4 fF	10fF	30fF	100 fF
0.010	24 ps	xxx ps	yyyy ps	zzz pss
0.03	xxxx	xxxxx	xxxxx	xxxxx
0.10	yyyy	yyyy	yyyy	yyy

Because you must know input slope, must also have lookup table for OUTPUT slope based on input slope and output load

Delay, Slope Lookup Tables

- When characterizing gates via lookup tables, must make sure that output loads represent the minimum/maximum loads in your designs
- Input slopes must represent minimum/maximum slopes in your designs
- Any input slopes, output loads that are outside table ranges must be clipped to min/max table values
- Actual input slopes/output loads will not match table axis values, so must interpolate between axis values using a 2D surface interpolation.