Graph Matching and Clock Gating

Prepared by Shmuel Wimer
Bar-Ilan Univ., Eng. Faculty

Outline

How data-driven clock gating works

Flip-flops activity and correlation

The optimal fan-out of a gater

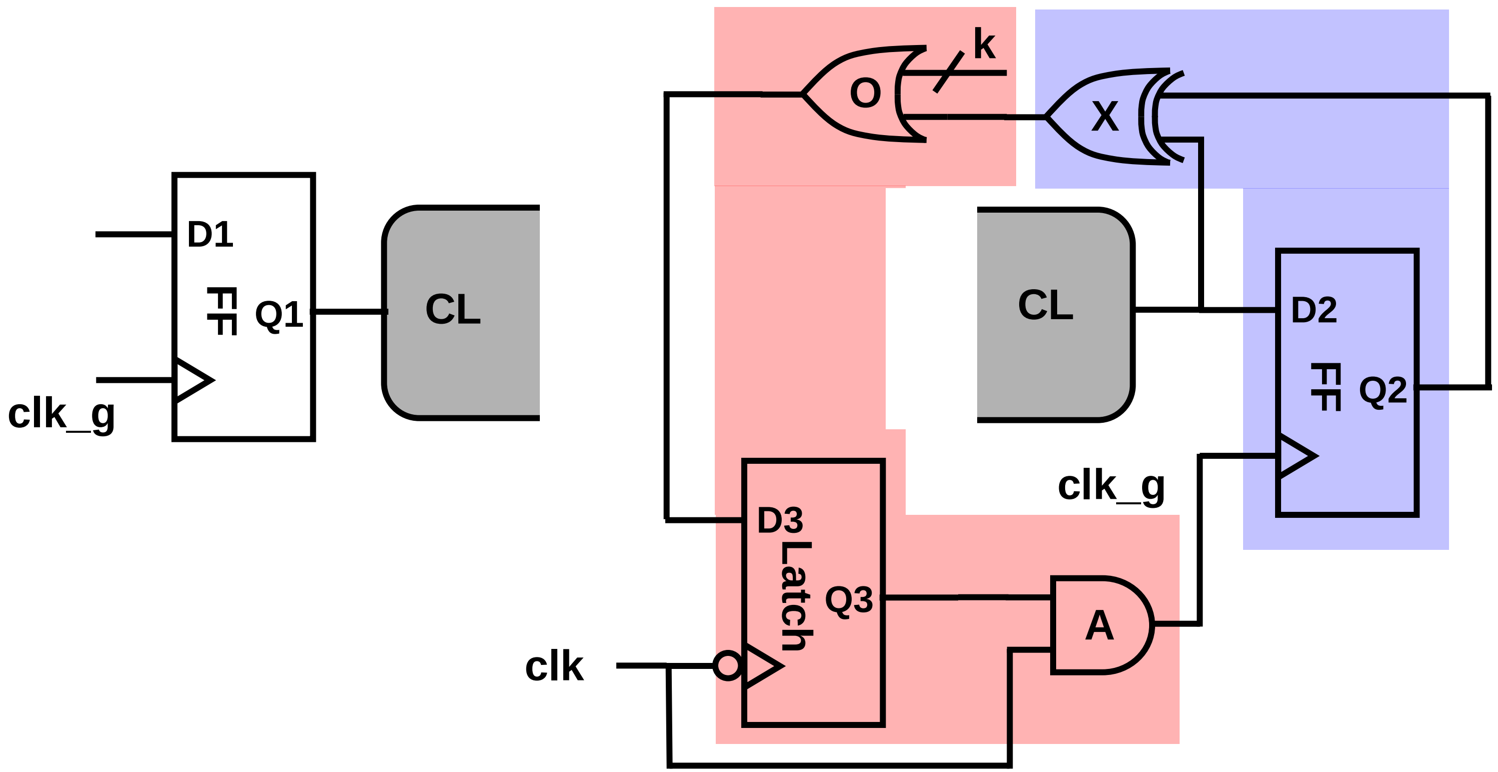
Delay implications

The complexity of flip-flop grouping

Motivation

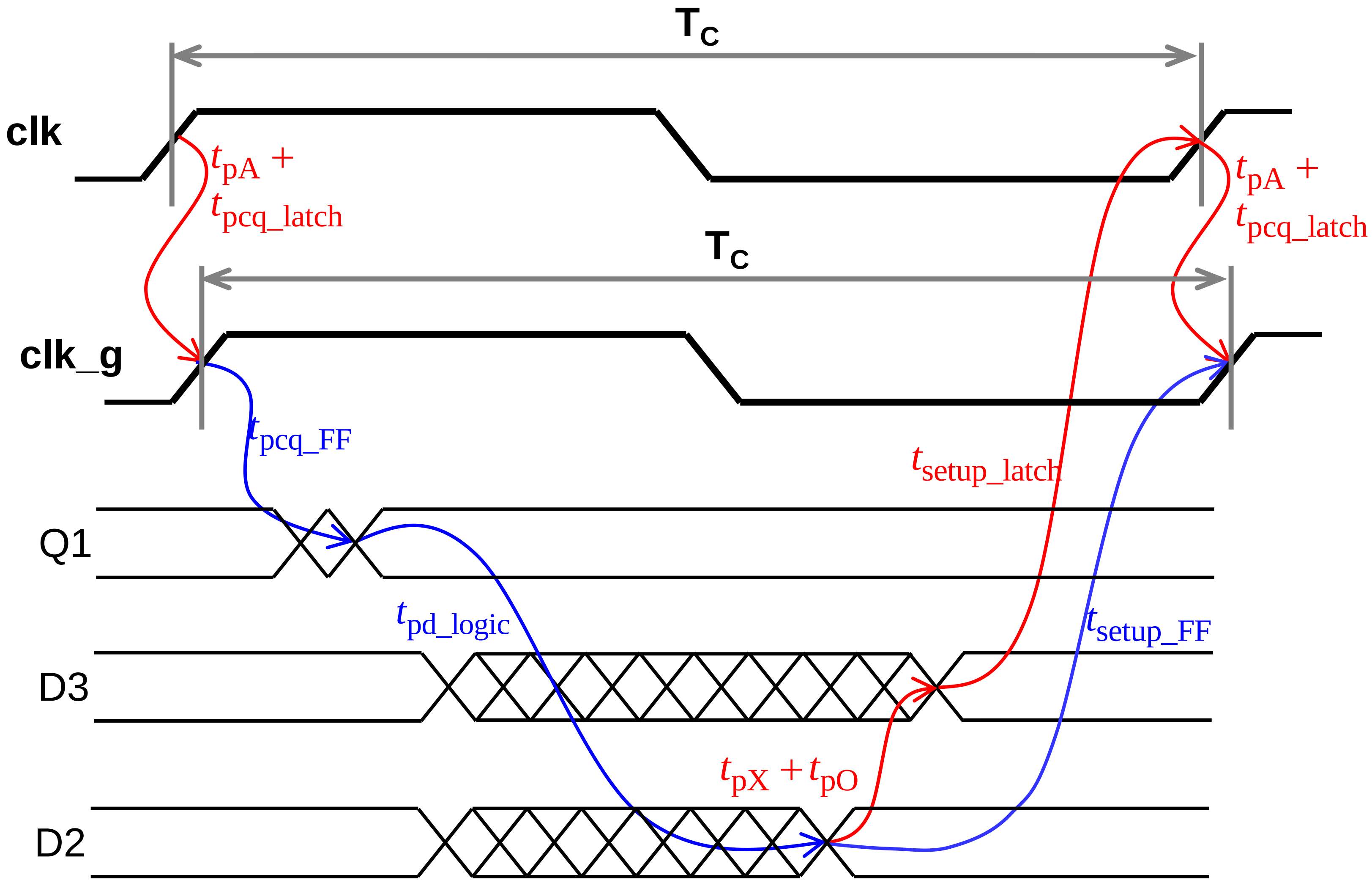
- Clocking consumes 30% to 70% of dynamic power
- Typically, only 3% of the clock pulses are useful, namely, only 3% of the clock switching occurs with data toggling!
- Clock enabling is easier at high design levels but harder in logic and gate level
- Clock enabling is easier in register files and data path, but harder in control
- Designers are conservative, leaving on table a lot of “hidden disabling”

Data-Driven CLK Gating



There is timing overhead!

Timing Implications ☹



The Optimal Clock Gater Fan-out

There is a tradeoff between hardware overhead and amount of saved clock pulses (power savings).

FFs' activities and their correlations is a key.

Worst case assumption:

All FF are toggling independently of each other.

k: # flip-flops, ***q***: FF probability for D=Q

Net saving at a leaf flip-flop

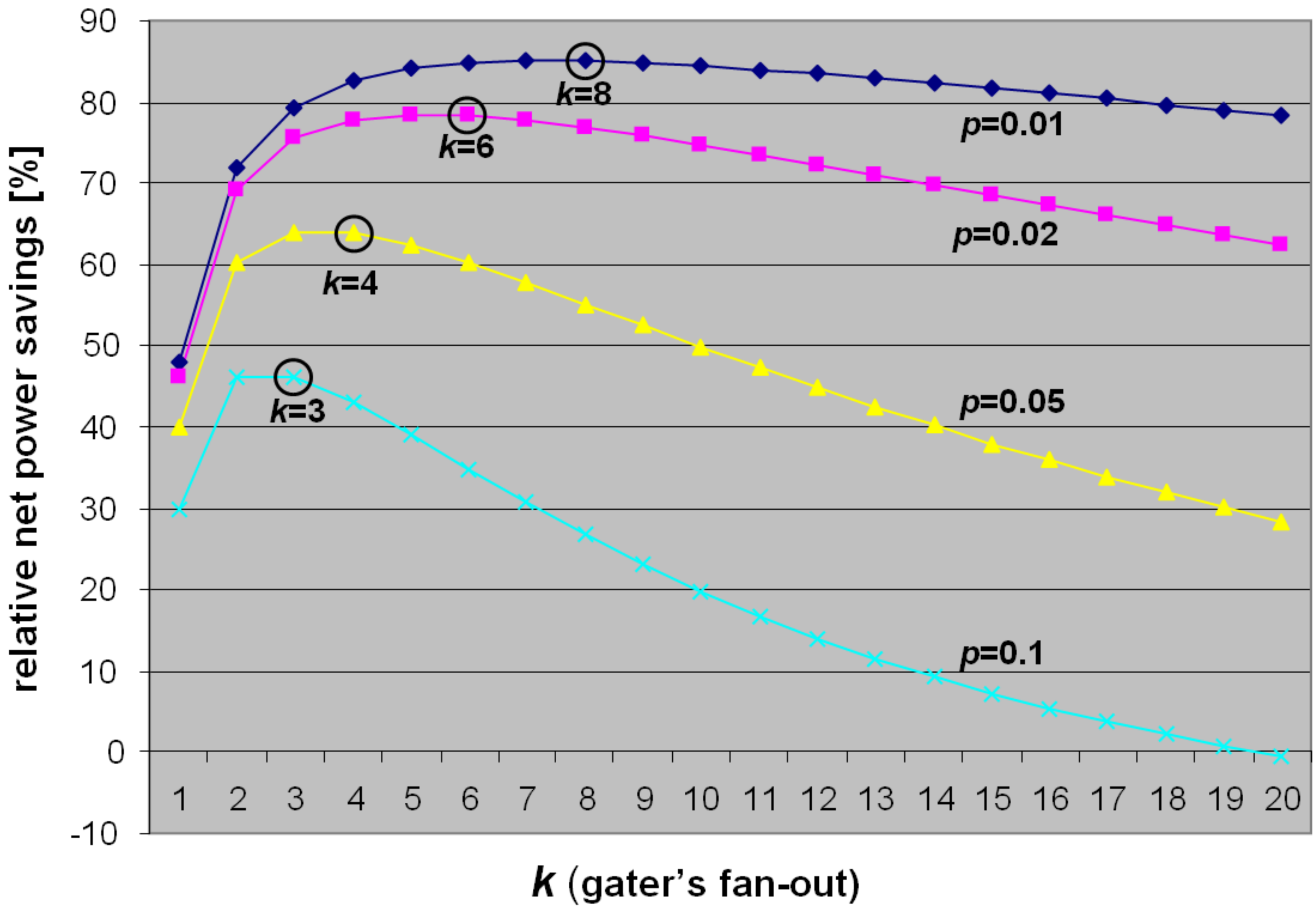
Latch overhead amortized over *k* FFs

$$c_{\text{net\_saving}} \geq q^k \left(c_{\text{FF}} + c_{\text{w}}\right) - \left[c_{\text{latch}}/k + \left(1-q\right)\left(c_{\text{w}} + c_{\text{OR}}\right)\right]$$

Gater’s disabling probability

Switching probability of FF enabling

Derivate by *k:* $$q^k \ln q \left(c_{\text{FF}} + c_{\text{W}}\right) + c_{\text{latch}}/k^2 = 0$$



90
80
70
60
50
40
30
20
10
0
-10
relative net power savings [%]
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
k (gater's fan-out)
k=8
k=6
k=4
k=3
p=0.01
p=0.02
p=0.05
p=0.1

Optimal *k*-size Flip-flop Grouping

Given n flip-flops and $m+1$ clock cycles

$\mathbf{a}=\left(a_1,\ldots,a_m\right)$ is the activity (toggling) of flip-flop

$\left\|\mathbf{a}_i \oplus \mathbf{a}_j\right\|$ is the number of redundant clock pulses ocurring by jointly clocking FF_i and FF_j

FF Pairwise Activity Model

$G(V,E,w)$: FF pairwise activity graph.

$v_i \in V$ corresponds to FF_i.

$e_{ij} = (v_i, v_j) \in E$ is FF pairing.

$\mathbf{a}_i \mid \mathbf{a}_j$ is joint toggling.

$w(e_{ij}) = \|\mathbf{a}_i \oplus \mathbf{a}_j\|$ is redundant clock pulses, hence a waste.

$E' \subset E$: vertex matching

Assume that FF grouping in pairs (k=2).

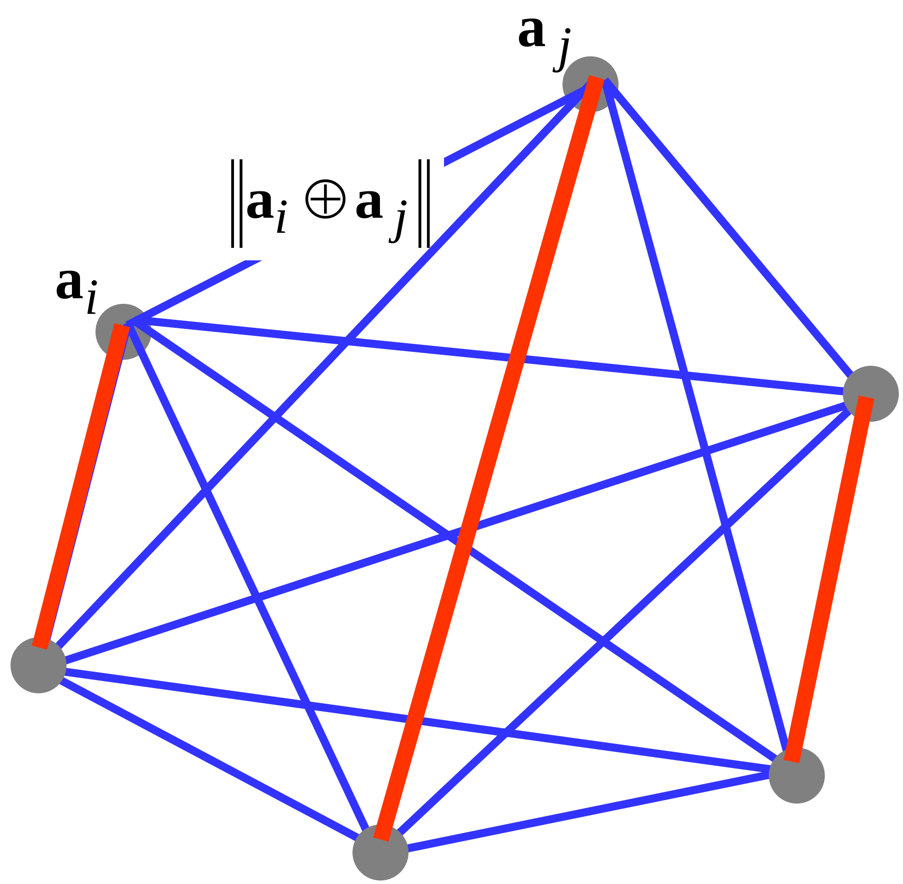
Total power, normalized to number of clock switching:

$$P = 2\sum_{e_{ij}\in E'} \left\| \mathbf{a}_i \mid \mathbf{a}_j \right\| =$$

$$\sum_{v_i\in V} \left\| \mathbf{a}_i \right\| + \sum_{e_{ij}\in E'} \left[\left\| \mathbf{a}_i \oplus \left(\mathbf{a}_i \mid \mathbf{a}_j \right) \right\| + \left\| \mathbf{a}_j \oplus \left(\mathbf{a}_i \mid \mathbf{a}_j \right) \right\| \right] =$$

Essential + Waste

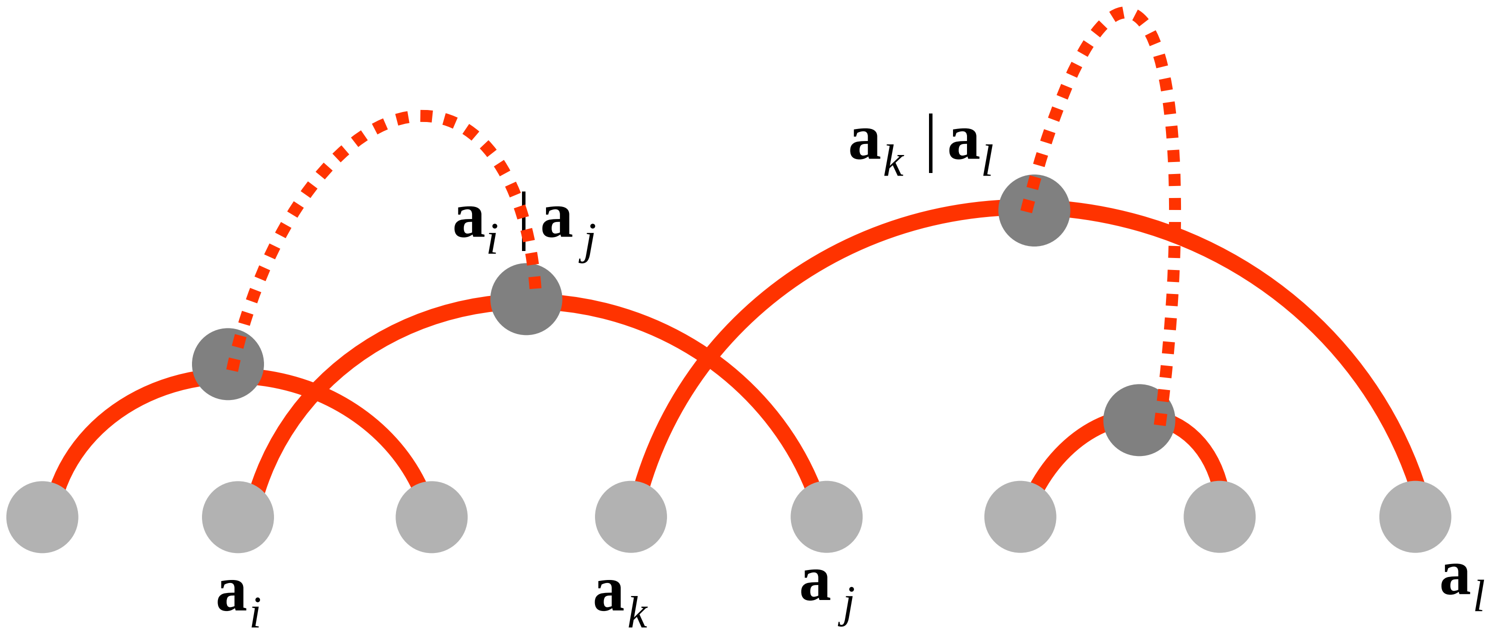
$$\sum_{v_i\in V} \left\| \mathbf{a}_i \right\| + \sum_{e_{ij}\in E'} \left\| \mathbf{a}_i \oplus \mathbf{a}_j \right\| = \sum_{v_i\in V} \left\| \mathbf{a}_i \right\| + \sum_{e_{ij}\in E'} w\left(e_{ij}\right)$$



Optimal FFs pairing ($k = 2$) is solved in polynomial time by minimal cost perfect graph matching.

FF1:	0	0	1	0	0	0	1	0	0	0	0	1
FF2:	0	1	0	0	0	1	1	0	1	1	0	1
FF3:	1	1	1	0	0	0	0	1	0	1	1	0
FF4:	1	0	1	0	0	0	0	1	1	1	1	0
FF5:	1	0	0	1	1	0	1	0	1	1	0	0
FF6:	1	0	1	1	1	0	1	0	1	0	0	1
FF7:	0	1	1	1	0	1	0	0	1	0	0	1
FF8:	0	1	1	1	1	0	0	1	0	0	0	0

What happens when $k>2$?



Is repeated perfect matching optimal ?

No! Here is the optimal 4-size grouping

FF1:	0	0	1	0	0	0	1	0	0	0	0	1
FF2:	0	1	0	0	0	1	1	0	1	1	0	1
FF6:	1	0	1	1	1	0	1	0	1	0	0	1
FF7:	0	1	1	1	0	1	0	0	1	0	0	1
FF3:	1	1	1	0	0	0	0	1	0	1	1	0
FF4:	1	0	1	0	0	0	0	1	1	1	1	0
FF5:	1	0	0	1	1	0	1	0	1	1	0	0
FF8:	0	1	1	1	1	0	0	1	0	0	0	0

Finding optimal *k*-grouping is NP-hard