RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Exercise:
PLL / VCO

Prof. Dr. P. Fischer

Lehrstuhl für Schaltungstechnik und Simulation
Uni Heidelberg

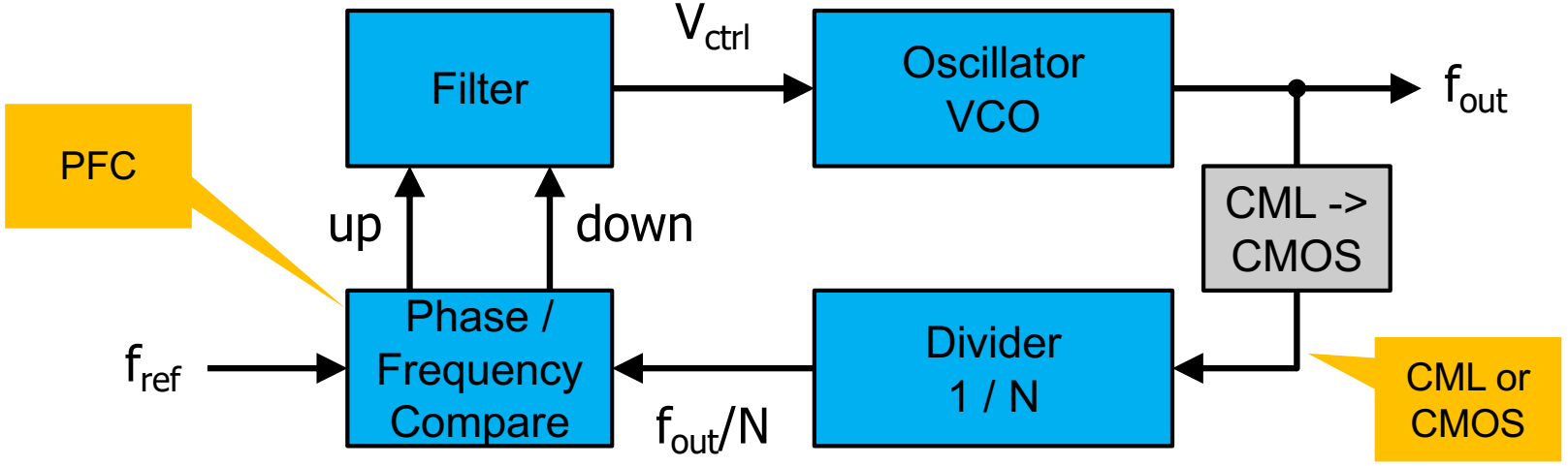
PLL

- In many applications, a very fast clock (some GHz) is required inside a chip.
- Such fast signals are hard to generate and distribute on PCBs etc.

- A ‘Phase Locked Loop’ = PLL is a building block which generates a high frequency clock internally
- The frequency of that clock is stabilized by comparing it to a (slower) reference clock (often from an external crystal oscillator or so).

- PLLs are also used to
  - reconstruct a clock signal from a serial data stream (CDR = clock – data recovery)
  - Generate a ‘good’ (low jitter) clock from a ‘bad’ clock. ‘Jitter Cleaner’

PLL Elements



- The fast clock is generated in a 'VCO' oscillator with variable frequency
- The output frequency f_{out} is divided by N and compared to an external reference f_{ref}.
- When f_{out}/N is slower than f_{ref}, speed of the VCO is increased
- Lock is obtained, when f_{out}/N and f_{ref} are identical, i.e. in phase
- The filter makes sure the feedback system remains stable.
- (Mathematical description and stability of the regulation is a bit tricky and not discussed here, see literature)

1. The VCO

- Design a Voltage Controlled Oscillator (VCO) based on a ring oscillator
- Chose a voltage controlled delay element
  - (Inverter with variable capacitive load)
  - (Gain stages with variable bias)
  - Differential stages with variable bias. Use clamping loads
- Aim at a frequency range of of 100 MHz … 2 GHz
- Assume you have a supply independent voltage or current to bias
- Try to obtain a good power supply rejection
- In simulation, best start up with a clear initial condition

2. The Divider

- When using differential logic for the VCO, we can
  - Convert to CMOS and do 1/n and PFC in CMOS
  - Use Current Mode Logic (CML) throughout

- I suggest to go for CML

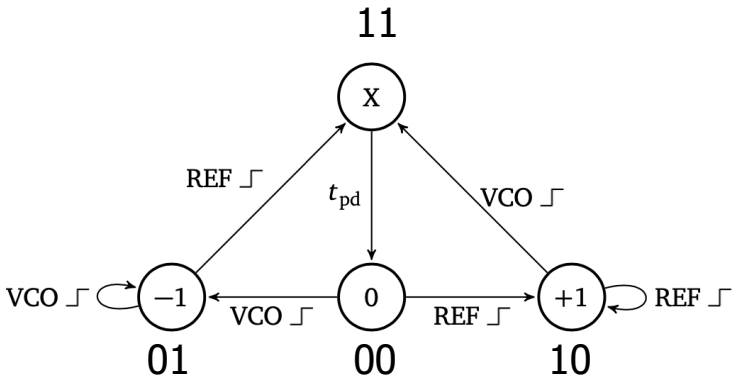
- Let us divide by, say, 8 for simplicity. Let us use 3 cascaded ’divide by 2’ toggle FFs

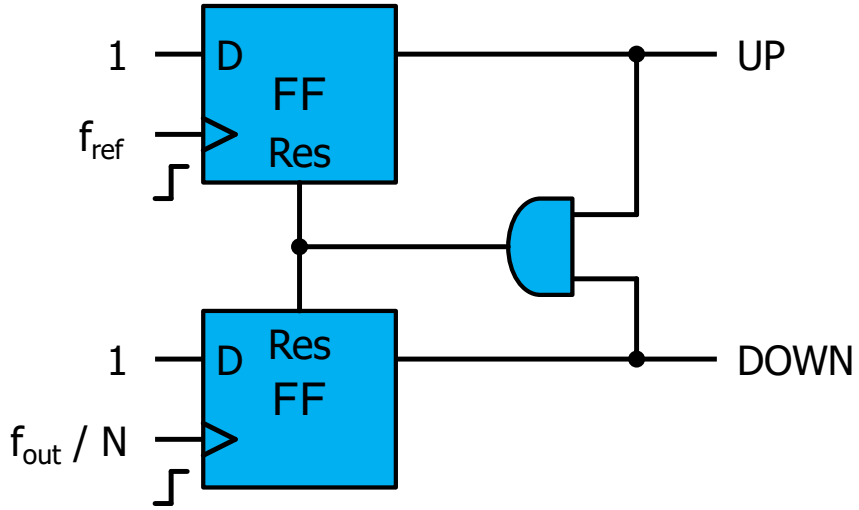
- Compose the CML flipflop from two CML latches, see ‘logic’ slides page 118. You do not really need a reset.

- Simulate the FF stand alone. Add another FF as (capacitive) load. How fast can you clock?

3. The PFC

- A very common PFC is shown below.
  - When both FFs are 1, they reset to 0 (asynchronous reset)
  - The clock edge which arrives ***first*** sets the Up ***or*** Down signal. This is reset back by the other edge.
- When one frequency is higher, the corresponding signal (UP/DOWN) is set more often.
- For equal frequencies, if f_{ref} is a bit earlier, UP will be set at every clock period, DOWN will stay quiet





3. The PFC

- Design a FF with (async.) reset.
  - The first latch can be simplified a bit because the D input is fix
- Simulate the PFC stand alone
  - Same input frequencies with phase shift
  - Slightly different frequencies

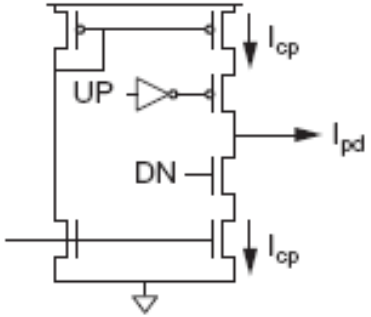
4. The Filter

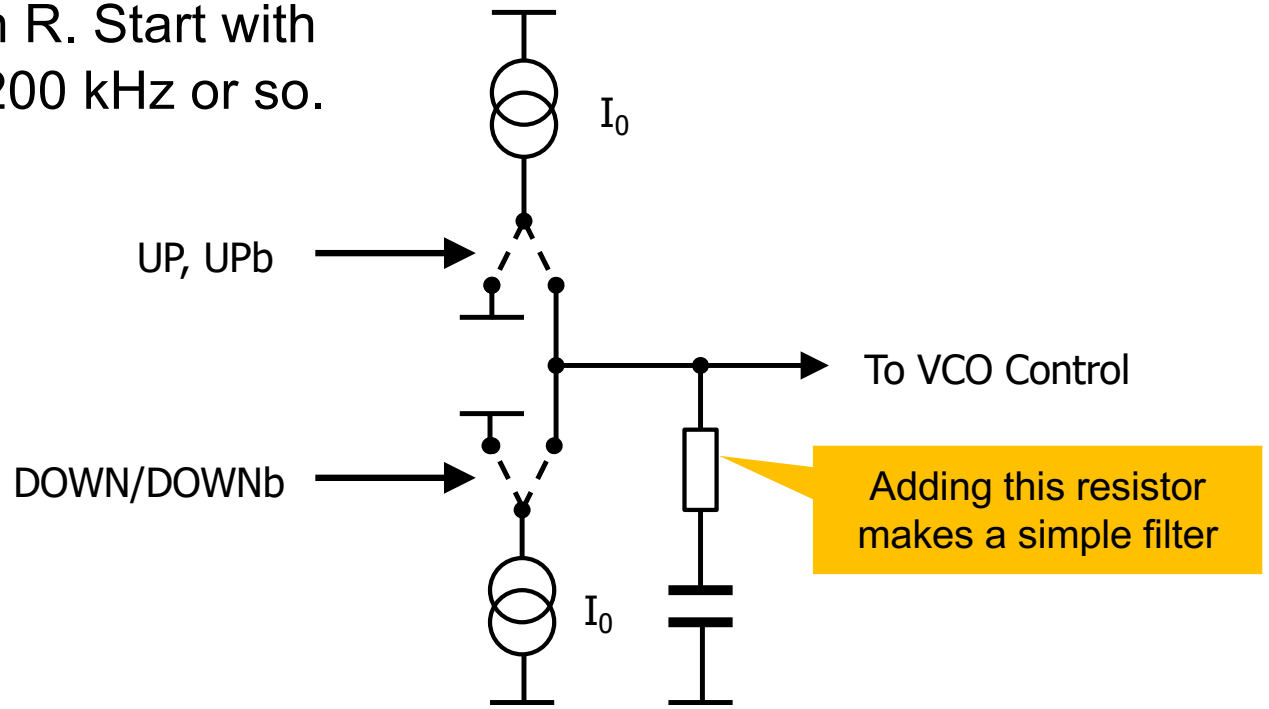
- The UP/DOWN signals must increase/decrease the control voltage of the VCO
- We use UP/DOWN to charge/discharge a capacitor.
  - This is called a ‘charge pump’

- To ensure stability, we need a ‘loop filter’. This can be incorporated in the charge pump
- Good filter design is tricky to find a compromise between
  - Stability
  - Fast lock (i.e. quickly follow changes on f_{ref}), if needed
  - Low phase noise (‘slow regulation’)
  - …

4. Charge Pump and Filter

- The (differential) Up/Down signals can be used to steer the current (check DC ranges! You may need mirrors!)
- Use the simple filter shown below. Use I_0 ~1uA.
  - Calculate a starting value for C for your I0: For – say – 50 'off-phase' clock periods, the control voltage changes enough to bring back the phase.
  - Play with R. Start with 1/RC = 200 kHz or so.





The PLL

- Put everything together.
- Set the initial condition of the filter cap such that you get a good frequency
- Set the reference frequency a bit different

- Does regulation work? If not, play with the filter RC.