Administrative stuff:
1. Project & final assignment due Thursday.
2. Final exam next Tuesday.
3. Notes on grading (see below).

Keep from last class:
1. Fractional-N PLLs
   - EA modulation for noise shaping.
   - Operation of 1st order, 1 bit EA modulator.
2. Dual loop PLLs
3. Clock and data recovery
   - General requirements
   - CE & NRZ data encoding.

Resume with the Hogg phase detector.

Grading:

I have received some feedback on the grading system I am using:
- non-standard grading, upset that people will get below 3.5.
- exams weighted too heavily, but more marks for effort.

Here is my justification:

The purpose of grading is to convey some information on how well you learned the material.
- If everyone gets a 4.0, no information is conveyed.
- Actual average for 500 level courses in EE is 3.3, I am using a 3.6.
- There is nothing non-standard about assigning a certain number of people to each GPA, that is how it is normally done, you are just not aware.
- Just transferring one distribution to another.

Effort is good, and I will factor that into my considerations, but you can't quantify it, so to do well in a course you have to know what you are doing, which is reflected in project & exam grades.

If there was a problem with the grading scheme it should have been voiced much earlier.

If you are unhappy with your final grades come see me, I will be happy to discuss it.
2 main types of phase detector for use in CDR PLLs:

1) Hogge Phase Detector
   - linear, can use PLL analysis we have already developed

2) Alexander (Bang-Bang) Phase Detector
   - non-linear, analysis can be difficult


A DFF produces a delayed version of its input
- use this with XOR gate to allow synchronous edge detection

First attempt:

![Diagram of a phase detector circuit]

Plot waveforms:

- Output $Y$ will have one pulse for each transition of the data
- Width of the pulse is linearly related to the phase offset between Data in and CLK.

Q: Will this work as a phase detector?

Problem: DC level of output depends on phase offset AND transition density
- we only want it to depend on the phase offset

Solution: Subtract reference pulses of fixed width on each transition.

![Diagram of a modified phase detector circuit]
- In this example, average value of the pulses is <0, control voltage will be reduced.

- Under locked condition, pulse width are equal.

Observation: 1) Detector gain depends on transition density.
- For design, we need to know the characteristics of the data stream we will be recovering.

2) Each transition on the data causes two pulses of width $T_{ck}/2$.
- If PLL is locked, these are equal.
- If these are used to drive a charge pump, each transition will be accompanied by a disturbance on the control voltage line, leading to data dependent jitter.
- Modifications to the standard Hegge architecture have been proposed to ameliorate this issue.

3) Finite delays in the FFs can cause problems at high frequencies.
- Assume each FF has a $CLK\rightarrow Q$ delay of $\Delta T$.
- If $B$ changes, $\Delta T$ seconds after $CLK$ rises, so the pulse on $Y$ will be $\Delta T$ seconds wider than it should be.
- For FF2, $B$ has already been delayed by $\Delta T$, so experiencing the same delay, so pulse on X will be exactly $T_{ck}/2$, with no extra width.
- $CLK$ will be offset from the optimum sampling time for the retimed data by $\Delta T$.
- At high frequencies, this can cause problems.

-Optimal sampling reducing the jitter tolerance.

Point: 1 - Circuitry have been proposed to correct this.
- Uses an early-late detection scheme.
- On each clock cycle, samples the incoming data stream at three points:
  - If no edge, no action taken.
  - If there is an edge:
    - If edge leads clock edge, generate fixed up pulse.
    - If edge lags clock edge, generate fixed down pulse.

Waveforms:

- **Clock Early:**
  \[ S_1 \neq S_2, \quad S_2 = S_3 \]
  \[ S_1 \oplus S_2 = 1, \quad S_2 \oplus S_3 = 0 \]

- **Clock Late:**
  \[ S_1 = S_2, \quad S_2 \neq S_3 \]
  \[ S_1 \oplus S_2 = 0, \quad S_2 \oplus S_3 = 1 \]

- **No Transition:**
  \[ S_1 = S_2 = S_3 \]
  \[ S_1 \oplus S_2 = 0, \quad S_2 \oplus S_3 = 0 \]

- From samples, XOR logic with \( S_1, S_2, S_3 \) can be used to generate output.

**Phase detector topology:**

![Phase detector diagram]

**Notes:**
- Sampling points are defined by FF1 & FF3; the other FFs merely serve as delay elements.
- FF3 is negative edge triggered.
- \( S_1 \) is also the re-timed data.
Waveform:

- Could also draw out waveforms for leading clock edges (left as an exercise).

- Since the P.D. output is a fixed pulse of one clock cycle duration, regardless of the actual magnitude of the phase error, the operation is non-linear with very high gain in the vicinity of $\Delta \phi = 0$.
  - This makes modelling difficult; it is best done through simulation.
  - The phase detector's high gain causes it to lock quickly (faster than Hegge Fig. 17, in general).

- Steady state jitter is dominated by bang-bang behavior.

**Limit cycle:** - When data transitions on every clock cycle, phase will alternate between leading and lagging the clock.

  - This is the steady state jitter mentioned above.

- Now consider the case where a delay $\tau$ is introduced between the sensed phase error $e_{kl}(t)$ and its effect on the VCO control voltage $v_{ctrl}$ (eg, due to loop filter).

  - New waveform: $v_{ctrl}$

```
+-------------------+-------------------+
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
```

```
+-------------------+-------------------+
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
|                   |                   |
```
The delay increases the period of the limit cycles, increasing the jitter.

- Approaches for counteracting delays have been proposed in the literature.

- At high data rates, it can be hard to implement phase detectors that operate fast enough.

- CDR circuits are often implemented in III-V semiconductors for their high-speed/lownoise properties.

- To get around this, consider reduced rate (i.e., 1/2 rate) phase detectors, where the VCO frequency (and thus the PLL frequency) is lower than the clock rate.

- We won’t cover these, but here is a reference if you are interested in an example of these phase detectors:


**CDR Performance Specifications:**

1. Jitter
   - Jitter generation
   - Jitter tolerance
2. Jitter peaking (in transfer function)
3. Acquisition time
   - e.g., 10 ns for SONET system
4. Bit error rate
   - e.g., $< 10^{-12}$ for SONET system

**PLLs for Indirect Modulation**

The function of a radio transmitter: translate a low frequency modulation signal to a desired RF band for efficient transmission over a wireless channel.

- Various architectures for achieving this goal.

- Traditionally, the most popular has been the Super-heterodyne architecture, proposed by Armstrong in 1918.
Superheterodyne transmitter:

- **Pros**: IF filter can be highly selective, filter requirements are relaxed compared to how they would be if done at RF.
- **Cons**: - multiple mixers, LOs
  - IF filtering must be done off-chip, requires matching to SAW, burns more power.

→ To achieve higher levels of integration, in recent years people have moved towards direct conversion transmitters.

**Direct Conversion Transmitter**:

- Only one mixer, now baseband is converted directly to RF.
  - Fewer off-chip components required.

→ For freq. & phase modulation schemes, we can eliminate mixers entirely by using direct modulation w/ a VCO.

**Direct modulation transmitter**:

- Channel Select.
  - PD
  - LPP
  - DIV
  - Modulation
PLL is required anyway to generate the LO in the other architectures.

Drawbacks:
- VCO characteristics are sensitive to temp. & process variations, so PLL is needed to set the frequency.
- PLL cannot influence VCO during transmission, so open loop approach is used.
- VCO is very sensitive to undesired perturbations.
  → Minimize leakage currents.
  → Active high isolation (exclude possibility of single-chip implementation - the original goal).
- VCO Vary: Freq. characteristics must be accurately modelled.

**Indirect Modulation Transmitter**

→ VCO now operates in closed-loop mode, so not as sensitive to disturbance or direct-modulation scheme.

→ This is the architecture we will be discussing.

- One option would be for Modulator to be 1 bit that controls a dual modulus divider.

**Time Domain**

<table>
<thead>
<tr>
<th>Data</th>
<th>Freq. Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

**Freq. Domain**

- Drawbacks: 1) freq. resolution for modulator limited. (Similar to discussion for freq. synthesizers)
  2) Inefficient use of spectrum (due to sharp transitions in frequency)
Solutions:

1) Use E\(\Delta\) modulated fractional-N PLL

2) Use pulse-shaping filter to eliminate sharp edges in data and yield a more compact spectrum.

Observations:

- Previously, we saw that changing the divider value is equivalent to changing the phase/freq. at the input.
- As the shaped modulation data experiences the closed-loop transfer function \(H(s)\) in being transferred to the output.
- To avoid distortion of the desired pulse shape, the PLL bandwidth must exceed the modulation bandwidth.

Recall from our discussion of E\(\Delta\) modulators in PLLs that the quantization noise is pushed to high frequencies.
- To filter out this noise, we must limit the PLL bandwidth.

For a given phase noise requirement, we are limited in our PLL Obs, and this in our module\(\Delta\) bandwidth.
- To increase the bandwidth, we can increase the order of the E\(\Delta\) (or the PLL) to put more noise to high frequencies.
- We can also increase the sample rate of the E\(\Delta\) (reference freq. of PLL).

To both of these add complexity and power consumption, we would like a solution that allows higher modulation bandwidths without these drawbacks.
Solution: Add a pre-emphasis filter.

Pre-emphasis filter pre-distorts the modulation data, making the PLL dynamics have a higher effective bandwidth.

- No change in power, minimal increase in complexity as the filtering is done in the digital domain.

Limitations:

1) Matching - Compensation filter is implemented in the digital domain and will not change, but PLL dynamics are based on analog parameters and can have significant PVT variation.

- Parallel to parasitic pole-zero pair that distorts the transmitted data and leads to increased bit-error-rate.

- PLL dynamics are no longer flat.

2) Dynamic Range - Clearly we cannot use this technique to obtain infinite bandwidth. What limits us here? 

- Dynamic range of the nodes internal to the PLL.

- Greater duty in fast divide ratio, need higher modulus divider.

- Greater duty in fast phase error, limited by constraint that PLL must remain locked (limitation imposed by P.I.D.)

* For more information, look up thesis 
  papers by Michael A. Perrott