PRESCALERS

Frequency Synthesizers

Using a reference counter in a PLL synthesizer

resolution: \( F_1 \)

J. A De Lima – Introdução ao Projeto de CI’s de Sinais Mistos CMOS
Synchronous vs. asynchronous counters (high-frequency)

Synchronous counters
- consume large power
- represent large CLOAD to oscillator
- race problems

Ex: if Q2 is slower to go to 0 than Q1 to go to 1, the output of the AND gate experiences glitches.
• delay almost constant between the input clock and the output at the divided frequency (NOT proportional to no. of flip-flops)
Divide-by-5 single prescaler

Dual 3/4 prescaler and state diagram

J. A De Lima – Introdução ao Projeto de CI’s de Sinais Mistos CMOS
asynchronous counters ⇒ preferred option for high-frequency

- forward/backward counting
- power consumption is reduced, as each stage operates at half frequency of previous stage

Asynchronous counters: (a) cascade of toggle flip-flops and (b) cascade of modulo-2 Johnson counters
• delay is however added between the input clock and the output at the divided frequency (proportional to no. of flip-flops)
Programmable dividers

Presettable modulo-$P$ asynchronous counter (modulus-8 with backwards counting from $Q3Q2Q1=111$ to 000)

Basic principle: to preset counter to an initial state $P$ and detect final state $F$ by means of an ‘end-of-count’ EOC logic $\Rightarrow$ the counter counts down between $P$ and $F$

- limitation is max $f_{in}$ as correct operation is guaranteed if EOC signal presets the counter before the next clock edge arrives

J. A De Lima – Introdução ao Projeto de CI’s de Sinais Mistos CMOS
i) High-frequency operation is attained when logic function is kept simple

i) simplest dividers divide by fixed numbers

⇒ programmable divider could have a fixed-modulus high-speed divider as first stage

• If a pre-settable modulus-P divider follows a modulus-N prescaler, overall frequency division ratio is NxP.

• the input frequency has to be lowered exactly by P to keep same resolution

⇒ implies narrowing the PLL loop bandwidth, which may be undesirable!
Basic prescaler

channel selection

resolution degraded: \( P \times F_1 \)
Pulse Swallowing Technique

• If *S input pulses are swallowed* the output period becomes longer by *S* reference periods

⇒ overall frequency division-ratio is \( M = (NP + S) \), *which can be varied in unity steps* by changing *S*

\[
M_1 = (NP + S) \\
M_2 = (NP + S + 1) \\
⇒ \Delta M = M_2 - M_1 = 1 \text{ (same resolution without prescaling!)}
\]
asynchronous counter

- at beginning of counting, $N+1$ factor is selected in dual-modulus prescaler (OUT = 1)
- $P$ and $S$ count in parallel, with $P > S$
- when $S$ overflows, set $= 1$ and OUT $\to 0$
- $N$ factor is selected in dual-modulus prescaler
- it remains like that until $P$ overflows and OUT $\to 1$
- cycle is restarted.

**total counts of Fout is a full F1 cycle:**

$$S \times (N+1) + (P-S)N$$

$$SN + S + PN - SN$$

$$PN + S = M$$

---

J. A De Lima – Introdução ao Projeto de CI’s de Sinais Mistos CMOS
total counts of $F_{out}$ is a full $F_1$ cycle:

\[ S \times (N+1) + (P-S)N \]

\[ SN + S + PN - SN \]

\[ PN + S = M \]
Ex 1 (using pulsing swallowing):
• frequency synthesizer in 2400–2480 MHz ISM band
• 1 MHz channel spacing $\Rightarrow$ division factor ($M = NP + S$) between 2400–2480

*Design steps:*

i) choice of the modulus $N$ (*dual-modulus prescaler*), $P$ (*program counter*) and $S$ (*swallow counter*), with $P > S$

ii) assume that *only S can vary* to simplify channel-select logic.

iii) make either $N$ or $N + 1$ *a power of two*

iv) choose $S$ as low as possible so that $P > S$ is a minimum
Assuming initially \( S \) varies between 1 and 81 (to cover 81 possible division ratios) then \( P > 81 \):

\[
N < \frac{M - S}{P} = \frac{2399}{81} = 29.6
\]

Choosing \( N = 16 \) \( \Rightarrow \) \[
P = \frac{M - S}{N} = \frac{2399}{16} = 149 \quad \text{and} \quad P > S
\]

*Design values for pulse swallow*er

<table>
<thead>
<tr>
<th>( N )</th>
<th>( P )</th>
<th>( S )</th>
<th>( M )</th>
<th>( P &gt; S )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>149</td>
<td>16</td>
<td>2400</td>
<td>TRUE</td>
</tr>
<tr>
<td>16</td>
<td>149</td>
<td>96</td>
<td>2480</td>
<td>TRUE</td>
</tr>
<tr>
<td>32</td>
<td>75</td>
<td>0</td>
<td>2400</td>
<td>TRUE</td>
</tr>
<tr>
<td>32</td>
<td>75</td>
<td>80</td>
<td>2480</td>
<td>FALSE</td>
</tr>
<tr>
<td>32</td>
<td>76</td>
<td>48</td>
<td>2480</td>
<td>TRUE</td>
</tr>
<tr>
<td>22</td>
<td>109</td>
<td>2</td>
<td>2400</td>
<td>TRUE</td>
</tr>
<tr>
<td>22</td>
<td>109</td>
<td>82</td>
<td>2480</td>
<td>TRUE</td>
</tr>
<tr>
<td>9</td>
<td>256</td>
<td>96</td>
<td>2400</td>
<td>TRUE</td>
</tr>
<tr>
<td>9</td>
<td>256</td>
<td>176</td>
<td>2480</td>
<td>TRUE</td>
</tr>
</tbody>
</table>
\[ M = (P \times N) + S \]

\[ M_{\text{MIN}} = (P_{\text{min}} \times N) + S_{\text{min}} \]
\[ = ((N + 1) \times N) + 1 \]
\[ = N^2 + N + 1 \]

\[ M_{\text{MAX}} = (P_{\text{max}} \times N) + S_{\text{max}} \]

\text{Pmax and Smax are determined by the size of P and S counters.}

\text{M}_{\text{MIN}}-\text{M}_{\text{MAX}}: \text{range over which it is possible to change N in discrete integer steps.}
Ex2: assume that prescaler is programmed to $N/N+1 = 32/33$
S counter: 6 bits means $S$ can be $2^6 - 1 = 63$
P counter: 13 bits means $P$ can be $2^{13} - 1 = 8191$

$M_{\text{MIN}} = N^2 + N + 1 = 1057$
$M_{\text{MAX}} = (P_{\text{max}} \times N) + S_{\text{max}}$
$= (8191 \times 32) + 63 = 262175$

If $F_1 = 10\text{KHz}$ and $P = 6000; S = 40$

$F_{\text{OUT\_MIN}} = 1.057\text{GHz}$
$F_{\text{OUT\_MAX}} = 2.62\text{GHz}$

$F_{\text{OUT}} = F_1 (PN + S) = 10\text{KHz} (6000 \times 32 + 40) = 1.92040\text{GHz}$

$F_{\text{OUT1}} = F_1 (PN + S + 1) = 10\text{KHz} (6000 \times 32 + 41) = 1.92041\text{GHz}$

$\Delta F_{\text{OUT}} = 10\text{KHz}$
(Differential) CMOS Current Mode Logic - (D)CML

- The main building block of the before-described counters is the D-type level-triggered latch

- CK swing has to be wide enough ($V_{TH} + V_{GO}$) to turn on pMOS. Since CK has a finite slope, this implies a certain delay before the latch is able to sense at CK transition

$\Rightarrow$ Differential CML for high-speed processing

J. A De Lima – Introdução ao Projeto de CI’s de Sinais Mistos CMOS
• CML is based on the use of differential stages
• tail current is switched between two branches by CK
• a regenerative pair holds the data when CK is low
• loads can be triode-operating or diode-connected PMOS.
Single-Ended vs Differential

Common-Mode disturbances disappear in the differential output

J. A De Lima – Introdução ao Projeto de CI’s de Sinais Mistos CMOS
• small $\Delta V_{in}$ already develops full $V_{OUT}$
CML x CMOS

Pros:
 i) reduced voltage swing ($V_{GO}$ against $V_{GO} + V_{TH}$ of CMOS stages) $\Rightarrow$ less delay before input sensing $\Rightarrow$ higher speed
 ii) current-steering operation: current drained from supply less variable
 iii) differential circuits are immune to coupled disturbances; they reject disturbances coming from substrate and power supply due to other blocks

Cons:
 i) larger area
 ii) 2 “wires” per signal
 iii) higher consumption
- Emitter-Coupled Logic (origin)
- Emitter-Coupled Logic

ECL input/output characteristics.
CML AND / NAND gates

AND

NAND

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>out</th>
<th>out-not</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>out</th>
<th>out-not</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
CML OR / NOR gates
Merged AND gate and D-type latch in CML logic