Synthesis Techniques for CMOS
Folded Source-Coupled Logic Circuits

By
Prathyusha Ganti (900064564)
Outline

- Introduction
- Conventional CMOS Inverters
- FSCL Inverters
- Synthesis of Combinational FSCL Gates
  - Series Gating
  - MUX-MIN
  - VEM
- Synthesis of Sequential FSCL Gates
- Experimental Results
- Conclusion
- References
Conventional CMOS static logic offers several performance advantages in purely Digital Applications, including zero static power dissipation, high packing density, wide noise margin, high operating frequencies, etc.

However, in high performance Mixed signal environments, its major disadvantage is the generation of large amount of Digital switching noise.
Fig. 1. (a) A 450-uA current spike flows from Vdd to Gnd when a state transition occurs for (b) a conventional static inverter implemented in a 2-pm p-well CMOS process. The spike amplitude is approximately independent of frequency.
Reducing Digital Switching Noise

- Several techniques are used including power supply filters, diffused guard bands, and separate analog and digital supply lines, bonding pads, bonding wires, etc.
Reducing Digital Switching Noise

- Important to *minimize noise generation* than managing the noise transmission.
- The digital switching noise generated by CMOS source coupled logic [FSCL] is typically two orders of magnitude smaller than in conventional logic.
- Selective use of FSCL reduces Switching Noise in high frequency Mixed-signal integrated systems.
FSCL Inverters

- (a) PMOS FSCL
- (b) NMOS FSCL
FSCL Inverters Operation

- Current Steering is the basic principle of operation.

- The output voltage is low only when current \((\text{Iss}_1)\) flows from the output node to Gnd.
FSCL Gates

Measured power-delay characteristics from nine 39-stage FSCL-PMOS ring oscillators integrated in a 2-um p-well CMOS process.
FSCL Gates

SPICE results predict that FSCL NMOS is twice as fast as FSCL PMOS. AVL = 1 V and Vdd = 5 V. The ring oscillators were formed using two input NAND gates.
FSCL Inverters

- The measured current spike in the Vdd supply was 5uA compared to 450 uA for a static inverter.

- The simulated power-delay characteristics suggest that a version of NMOS diode loads is comparable in speed to conventional static logic.
Synthesis of Combinational FSCL Gates

- Series Gating Synthesis
- Multiplexer Minimization
- Variable-Entered Mapping Minimization
FSCL gate is comprised of two sections:

- A decision tree formed by a stack of NMOS differential pairs,
- And complementary output stages.
- The input differential tree is configured so that $I_{ss1}$ is steered to node $Qc$, only when the input vector $X_i = \{x_1, x_2, \ldots \}$ is “true” for $Qc$.
- $I_{ss1}$ is steered from node $Q$ to Gnd only when $X_i’$ is true for $Q$ (Xi is false for $Qc$)
Series Gating Synthesis

Fig. 4. In a p-well CMOS FSCL implementation, a function and its complement are synthesized in an NMOS decision tree. The output stages are identical for all gates.
Series Gating Synthesis

Fig. 5. A three-input NAND/AND FSCL gate synthesized using the series-gating technique. There are $2^K$ branches, $2^k - 1$ differential pairs, and $2(2^K - 1)$ devices in the input tree for a $K$-variable function.
Series Gating Synthesis

- The non-minimum series-gated realization of a K-variable logic function in FSCL requires $2^K$ branches, $2^K - 1$ differential pairs, and $2(2^K - 1)$ NMOS switching transistors.
Multiplexer Minimization method

- Reduces number of switching devices and number of branches.
- Based on Shannon Expansion theorem.
- Logic function $F(A, B, C, \ldots)$ with $n$-input variables is factored first with respect to the variable $A$ as
  
  $$F(A, B, C, \ldots) = F(0, B, \ldots) A' + F(1, B, \ldots) A$$
  
  \[\text{--- (1)}\]
Multiplexer Minimization method

- Factorization of the sub-functions with respect to the variable $B$ gives

\[ F(0, B, C, \ldots) = F(0, 0, C, \ldots) B' + F(0, 1, C, \ldots)B \]  
\[ -(2) \]

\[ F(1, B, C, \ldots) = F(1, 0, C, \ldots) B' + F(1, 1, C, \ldots)B \]  
\[ -(3) \]
Multiplexer Minimization method

- The MUX-MIN expansion of $I_c[3 \text{ i/p NAND/AND}]$ with respect to $A$ follows from (1) as

$$I_c(A, B, C) = ABC = I_c(0, B, C)A' + I_c(1, B, C)A$$
$$= OA' + (BC)A = ABC. \quad - - - \ (4)$$

Similarly, the expression for $I(A, B, C) = I'c(A, B, C)$ flowing into the tree from $Q$ is

$$I(A, B, C) = (ABC)' = A' + B' + C'$$
$$= I(0, B, C)A' + I(1, B, C)A$$
$$= A' + I(1, B, C)A \quad - - - \ (5) \quad \text{where}$$

$$I(1, B, C) = B' + C' = I(1, 0, C)B' + I(1, 1, C)B$$
$$= B' + (C')B = B' + BC'. \quad - - \ (6)$$
Multiplexer Minimization method

- When the input variables are ordered as
  \[ A \rightarrow B \rightarrow C, \]

  the final expressions resulting from MUX-MIN are

  \[ Ic(A, B, C) = ABC \quad \text{(7)} \]

  and

  \[ I(A, B, C) = A' + AB' + ABC' \quad \text{(8)} \]
Multiplexed Minimization Method

Fig. 6. A three-input FSCL NAND/AND gate after minimization.
Variable- Entered Mapping Minimization

- It is a technique used to minimize conventional multiplexer implementations of logic functions.

- VEM (Variable- Entered Mapping) is applied to FSCL to reduce the input decision tree.
Variable- Entered Mapping Minimization

- Construct an ordered truth table for Ic and I (corresponding to Qc and Q).
- The input variable associated with the lowest level is placed into the leftmost column followed to the right by variables corresponding to increasingly higher I-MUX levels.
Variable- Entered Mapping Minimization

- VEM is first applied to eliminate the highest level variable resulting in a second truth table, and then again to eliminate the second highest variable giving the third truth table, and so on.

- Minimization of a K-variable logic function requires K-1 recursions of the VEM technique.
Variable- Entered Mapping Minimization

Fig. 7. Ordered truth tables for the three-input NAND/AND functions: (a) initial table, (b) table after VEM on variable $C$, and (c) the final table after VEM on variable $B$. 
Examples with Different Ordering

- Different ordering results in different minimum trees.
- For a $K$-variable logic function, there are $K!$ possible orderings and $K!$ possible trees.
- As the number of trees differ in the number of switching devices, which affects the die area and in the number of branches which impacts the propagation delay and switching noise.
- Hence, variable order is chosen, that best equalizes the number of branches connected to the differential output nodes.
Examples with Different Ordering

<table>
<thead>
<tr>
<th>Input Variable Ordering</th>
<th>$I_C$</th>
<th>$I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ABC]</td>
<td>$(A' + AB')C + AB$</td>
<td>$(A' + AB')C'$</td>
</tr>
<tr>
<td>[ACB]</td>
<td>$A'C + AC + ABC'$</td>
<td>$A'C + AB'C'$</td>
</tr>
<tr>
<td>[BAC]</td>
<td>$(B' + A'B')C + AB$</td>
<td>$(B' + A'B')C'$</td>
</tr>
<tr>
<td>[BCA]</td>
<td>$B'C + BC + ABC'$</td>
<td>$B'C' + A'BC'$</td>
</tr>
<tr>
<td>[CAB]</td>
<td>$ABC' + C$</td>
<td>$A'C' + AB'C'$</td>
</tr>
<tr>
<td>[CBA]</td>
<td>$ABC' + C$</td>
<td>$B'C' + A'BC'$</td>
</tr>
</tbody>
</table>
# Examples with Different Ordering

**TABLE II**

<table>
<thead>
<tr>
<th>Input Variable Ordering</th>
<th># Devices</th>
<th># Levels</th>
<th># Branches at $Q_c$</th>
<th># Branches at $Q$</th>
<th>Delay (ns)</th>
<th>Spike ($\mu$A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ABC]</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>1.8</td>
<td>7.2</td>
</tr>
<tr>
<td>[ACB]</td>
<td>8</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3.2</td>
<td>7.1</td>
</tr>
<tr>
<td>[CBA]</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2.2</td>
<td>6.8</td>
</tr>
</tbody>
</table>
FSCL 1-b Full Adder

Fig. 9. FSCL (a) SUM and (b) CARRY circuits for a 1-b full adder.
Comparison of characteristics of Static CMOS and FSCL inverters and Full adders

<table>
<thead>
<tr>
<th>Property</th>
<th>Device</th>
<th># of Devices (PMOS/NMOS)</th>
<th>Device Size</th>
<th>Average Power</th>
<th>Average Delay</th>
<th>Power/Delay Product</th>
<th>Maximum Current Spike Per Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Static CMOS Inverter</td>
<td>2 (1/1)</td>
<td>((W/L)_N = 4/2)</td>
<td>0.86 mW @500 MHz</td>
<td>0.35 ns</td>
<td>0.3 pJ</td>
<td>0.7 mA</td>
</tr>
<tr>
<td></td>
<td>FSCL-NMOS Inverter</td>
<td>7 (2/5)</td>
<td>((W/L)_{1,2} = 8/2)</td>
<td>1.0 mW @500 MHz</td>
<td>0.4 ns</td>
<td>0.4 pJ</td>
<td>6 μA</td>
</tr>
<tr>
<td></td>
<td>Static CMOS Full Adder</td>
<td>36 (18/18)</td>
<td>((W/L)_N = 4/2)</td>
<td>0.9 mW @50 MHz</td>
<td>12 ns</td>
<td>11 pJ</td>
<td>1.85 mA</td>
</tr>
<tr>
<td></td>
<td>FSCL NMOS Full Adder</td>
<td>28 (2/24)</td>
<td>((W/L)_N = 8/2)</td>
<td>1.0 mW @50 MHz</td>
<td>3.0 ns</td>
<td>3.0 pJ</td>
<td>15 μA</td>
</tr>
</tbody>
</table>
Synthesis of Sequential FSCL Gates

- MUX-MIN and VEM methods are applied to the synthesis and minimization of sequential circuits.
NOR SR latch

(a) A nor SR latch and truth tables for VEM minimization. (b) A single-gate FSCL SR latch.
Complexity of FSCL

- FSCL is more complex for simple logic functions. However, it is significantly simpler for complex functions.
- For example, an FSCL inverter requires seven devices compared to two devices in static logic. And, a combined D-latch requires 17 devices compared to 28 in a conventional implementation.
Experimental Results

<table>
<thead>
<tr>
<th>Circuit</th>
<th># of Devices in FSCL</th>
<th># of Devices in CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Cell</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>SR Latch</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>D Latch</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>D Latch (Preset &amp; Clear)</td>
<td>13</td>
<td>18</td>
</tr>
<tr>
<td>D Latch with 2-to-1 MUX as D input</td>
<td>17</td>
<td>28</td>
</tr>
<tr>
<td>Full Adder</td>
<td>28</td>
<td>36</td>
</tr>
<tr>
<td>AND-OR</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>2-to-1 MUX</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>
### Experimental Results

#### TABLE V
**Measured Characteristics for Three-Input NAND and AND Gates and D-Type Flip-Flops Integrated in a 2-μm p-Well CMOS Technology**

<table>
<thead>
<tr>
<th>CK (MHz)</th>
<th>Power = 0.5 mW/Gate</th>
<th>Power = 1.0 mW/Gate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_{pair}$</td>
<td>$V_{dd}$ Spike</td>
</tr>
<tr>
<td>3 I/P NAND</td>
<td>—</td>
<td>5.9 ns</td>
</tr>
<tr>
<td>D-Latch</td>
<td>20</td>
<td>6.0 ns</td>
</tr>
<tr>
<td>D-Latch (PR/CLR)</td>
<td>20</td>
<td>6.4 ns</td>
</tr>
</tbody>
</table>
Conclusion

- FSCL features very low supply current spikes compared to conventional logic, and therefore, its selective use provides performance advantages in the design of high-frequency mixed-mode IC’s.
- The series-gating technique provides a direct, but not minimal implementation of logic functions in FSCL. The multiplexer-minimization or variable-entered mapping methods provide single-gate realizations with minimum generation of digital switching noise.
- FSCL implementations often provide a considerable savings in die area and power consumption for complex high-speed functions.
References

- Synthesis Techniques for CMOS Folded Source-Coupled Logic Circuits, Sailesh R. Maskai, Sayfe Kiaei, Member, IEEE, and David J. Allstot, Fellow, IEEE
  
  http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=148324

- Analog Logic Techniques Steer Around the Noise, David J. Allstot, Sayfe Kiaei, and Rajesh H. Zele
  
  http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=232788&userType=inst
Questions ??