Admin

- Homework due today. New assignment on your way.
- Should have received feedback on projects. Please submit revised abstract by Monday morning.
Pass-Transistor Logic

- N transistors
- No static consumption
- Transistor implementation using NMOS

Pass-Transistor Logic

- Performance of PTL:
  - Advantage over CMOS in implementing XOR, MUX
  - Disadvantage in implementing AND, OR.
- Datapaths, arithmetic circuits are examples of use:
  - Adders and multipliers use XOR, MUX
  - Advantage of complementary implementation
- Comparisons:
  - When a new logic family is introduced, the examples are chosen to show its advantages; (not disadvantages).
  - Comparison papers sometimes point to the disadvantages
- Full-custom design
Examples of PTL Styles

- Complementary Pass-Transistor Logic
  - NMOS-only pass-transistor network
- Transmission-gate logic
  - NMOS+PMOS pass gates
- Double Pass-Transistor Logic
  - NMOS+PMOS network
- Numerous other logic families

NMOS-only switch

Threshold voltage loss causes static power consumption
Solutions

- Transmission gates – adding complexity
- Low-threshold switches – leakage!
- Level-restoration

![Level Restorer Diagram]

Single-Ended Level Restoring

![Single-Ended Level Restoring Diagram]
Differential Level Restoring

Different level restoration leads to different logic families

Different Restoration Schemes

Swing-Restored Pass-Transistor Logic

Parameswar, et al
CICC’94, JSSC 6/96
Other Level-Restoring Schemes

Energy Economized Pass-Transistor Logic

DCVS with Pass Gates (DCVS-PG)

Pass-Transistor Logic Families

CPL

SRPL

DPL

DCVSPG
CPL Complementary Pass-Transistor Logic (CPL)

- Complementary functions
- Reduced number of logic levels
- Less transistors than CMOS
- Fast – reduced load
- Complementary inputs – complementary outputs
- $V_T$ drop – several solutions

Yano et al., CICC’89, JSSC 4/90

CPL

Same topology of networks
Just different signal arrangements
Complementary Pass-Transistor Logic (CPL)

- Fast
- $V_T$ drop
- Efficient implementation of arithmetic

CPL Karnaugh Maps
CPL vs. CMOS

<table>
<thead>
<tr>
<th></th>
<th>CMOS</th>
<th>CPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Adder Circuit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transistor Count</td>
<td>40</td>
<td>28</td>
</tr>
<tr>
<td>Area</td>
<td>4730 $\mu$m²</td>
<td>4216 $\mu$m²</td>
</tr>
<tr>
<td>Delay (4V)</td>
<td>0.63 ns</td>
<td>0.26 ns</td>
</tr>
<tr>
<td>Power (100MHz)</td>
<td>1.2mW</td>
<td>0.86mW</td>
</tr>
</tbody>
</table>

Skewing Output Inverter

![Graph showing Full-Adder Delay Time vs. CMOS-Inverter Delay Time](image)
## Differential vs. Single-Ended

<table>
<thead>
<tr>
<th></th>
<th>Single-Ended Pass Transistor Logic</th>
<th>CPL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CLA Circuit</strong></td>
<td><img src="image1" alt="Single-Ended CLA Circuit" /></td>
<td><img src="image2" alt="CPL CLA Circuit" /></td>
</tr>
<tr>
<td><strong>Transistor Count</strong></td>
<td>82</td>
<td>86</td>
</tr>
<tr>
<td><strong>Delay (t_{pd})</strong></td>
<td>0.26 ns</td>
<td>0.15 ns</td>
</tr>
</tbody>
</table>

## Double Pass-Transistor Logic (DPL)

### AND/NAND

![AND/NAND Diagram](image3)

### XOR/XNOR

![XOR/XNOR Diagram](image4)
Comparison of Logic Styles

Zimmermann, Fichtner, JSSC 7/97
## Comparison of Logic Styles

![Logic Styles Diagram]

## Results

### Full-Adder Comparisons

<table>
<thead>
<tr>
<th>Gate Type</th>
<th>Logic Style</th>
<th>Delay (ns)</th>
<th>Power (μW)</th>
<th>Size (λ²)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3.3 V</td>
<td>1.5 V</td>
<td>3.3 V</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with layout</td>
<td>extracted capacitances</td>
<td>without layout</td>
</tr>
<tr>
<td>FA</td>
<td>CMOS</td>
<td>1.89</td>
<td>7.88</td>
<td>1.11</td>
</tr>
<tr>
<td>CMPL</td>
<td>1.39</td>
<td>5.55</td>
<td>1.23</td>
<td>7.95</td>
</tr>
</tbody>
</table>

---

---
### Results

#### Logic Gates Comparisons (CMOS and CPL)

<table>
<thead>
<tr>
<th>gate type</th>
<th>logic style</th>
<th>delay (ns)</th>
<th>power (μW)</th>
<th>PT (norm.)</th>
<th># trans.</th>
<th>size (λ^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3.3 V</td>
<td>1.5 V</td>
<td>3.3 V</td>
<td>1.5 V</td>
<td>3.3 V</td>
</tr>
<tr>
<td>NAND2</td>
<td>CMOS</td>
<td>0.91</td>
<td>3.20</td>
<td>1.3</td>
<td>1.10</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>1.28</td>
<td>3.67</td>
<td>1.80</td>
<td>1.90</td>
<td>1.80</td>
</tr>
<tr>
<td>AND4</td>
<td>CMOS</td>
<td>1.30</td>
<td>5.28</td>
<td>1.90</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CMOS+</td>
<td>1.15</td>
<td>4.81</td>
<td>1.90</td>
<td>0.88</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>2.30</td>
<td>11.98</td>
<td>4.60</td>
<td>4.63</td>
<td>5.20</td>
</tr>
<tr>
<td>AOI/NOT1</td>
<td>CMOS</td>
<td>1.12</td>
<td>4.40</td>
<td>1.70</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>1.47</td>
<td>7.42</td>
<td>4.10</td>
<td>3.09</td>
<td>4.10</td>
</tr>
<tr>
<td>MUX2</td>
<td>CMOS</td>
<td>1.13</td>
<td>4.17</td>
<td>2.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CMOS+</td>
<td>1.59</td>
<td>6.50</td>
<td>1.90</td>
<td>1.37</td>
<td>1.50</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>1.28</td>
<td>6.21</td>
<td>1.90</td>
<td>2.03</td>
<td>2.54</td>
</tr>
<tr>
<td>MUX4</td>
<td>CMOS</td>
<td>2.03</td>
<td>7.50</td>
<td>2.60</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CMOS+</td>
<td>2.33</td>
<td>10.17</td>
<td>4.40</td>
<td>1.14</td>
<td>1.31</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>1.76</td>
<td>8.51</td>
<td>4.00</td>
<td>1.41</td>
<td>1.77</td>
</tr>
<tr>
<td>XOR</td>
<td>CMOS</td>
<td>1.43</td>
<td>5.51</td>
<td>2.10</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td>CMOS+</td>
<td>1.82</td>
<td>7.94</td>
<td>2.00</td>
<td>1.19</td>
<td>1.38</td>
</tr>
<tr>
<td></td>
<td>CPL</td>
<td>1.35</td>
<td>6.21</td>
<td>3.50</td>
<td>1.62</td>
<td>1.90</td>
</tr>
<tr>
<td></td>
<td>WANG</td>
<td>1.45</td>
<td>5.51</td>
<td>27.7</td>
<td>2.45</td>
<td>6.00</td>
</tr>
</tbody>
</table>

1. 2-input NAND/NOR combination (decomposition)
2. relaxed cell layout rules due to large number of (otherwise area dominating) input/output wires
3. does not work for $V_{dd} < V_{in} + |V_{th}|$

---

### Results

#### 32-bit Adder Comparisons

<table>
<thead>
<tr>
<th>logic style</th>
<th>delay (ns)</th>
<th>power (mW)</th>
<th>PT (norm.)</th>
<th>$I_{ss}/I_{sat}$ (%)</th>
<th># trans.</th>
<th># nodes</th>
<th>switching activity (%)</th>
<th>voltage (V)</th>
<th>process technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>4.14</td>
<td>7.50</td>
<td>1.00</td>
<td>23.6</td>
<td>1.60</td>
<td>1.60</td>
<td>1.40</td>
<td>2.8</td>
<td>0.5 μm</td>
</tr>
<tr>
<td>CPL</td>
<td>5.47</td>
<td>25.90</td>
<td>2.89</td>
<td>31.2</td>
<td>2.774</td>
<td>2.774</td>
<td>2.774</td>
<td>2.8</td>
<td>0.5 μm</td>
</tr>
<tr>
<td>CPL+</td>
<td>4.73</td>
<td>16.80</td>
<td>2.96</td>
<td>27.2</td>
<td>2.774</td>
<td>2.774</td>
<td>2.774</td>
<td>2.8</td>
<td>0.5 μm</td>
</tr>
<tr>
<td>DPL</td>
<td>5.00</td>
<td>15.00</td>
<td>2.42</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.3</td>
<td>0.5 μm</td>
</tr>
</tbody>
</table>

1. down-sized transistors
2. conditional sum adder in DPL, from literature [30]
Dynamic Logic

2 phase operation:
- Precharge
- Evaluation
- N + 1 Transistors
- Ratioless
- No Static Power Consumption
- Noise Margins small (NM)
- Requires Clock

Dynamic Gates

NMOS Inverter
PMOS Inverter

See Bowhill, Chapter 7.

Dynamic Logic

- **Advantages:**
  - Fast
  - Compact

- **Disadvantages:**
  - Less robust (noise margins, sensitive to leakage, noise coupling, charge sharing)
  - Needs clock

Logical Effort

\[ \text{LE} = \frac{\text{In}}{\text{Out}} \]
Logical Effort

\[ \text{LE} = \phi \]

\[ \text{Out} \]

Charge Leakage

\[ I_{\text{Leak}} = (I_{N_{\text{sub}}} + I_{N_{\text{diode}}}) - (I_{P_{\text{sub}}} + I_{P_{\text{diode}}}) \]

Time to switch the next gate: \[ t_{\text{sw}} = (C_{\text{DYN}} \cdot V_{\text{sw}})/I_{\text{Leak}} \]

Limits the minimum frequency: \[ f_{\text{min}} = 1/(t_{\text{sw}} \cdot \# \text{phases per clk cycle}) \]

Compensating Leakage

\[ V_{DD} \]

\[ \phi \]

\[ I_{BL} \]

\[ \text{(a)} \]

\[ \phi \]

\[ I_{BL} \]

\[ \text{(b)} \]

Charge Sharing (Redistribution)

\[ V_{DD} \]

\[ \phi \]

\[ V_{DD} \]

\[ M_p \]

\[ C_L \]

\[ \text{Out} \]

\[ M_a \]

\[ V_{out}^{(t)} \]

\[ A \]

\[ V_{DD} - V_{out}^{(t)} \]

\[ B = 0 \]

\[ M_b \]

\[ C_a \]

\[ \phi \]

\[ M_f \]

\[ C_b \]

\[ \Delta V_{out} < V_{Tn} \]

\[ C_L V_{DD} = C_L V_{out}^{(t)} + C_a (V_{DD} - V_{Tn}(V_X)) \]

\[ \text{or} \]

\[ \Delta V_{out} = V_{out}^{(t)} - V_{DD} = -\frac{C_a}{C_L} (V_{DD} - V_{Tn}(V_X)) \]

\[ \Delta V_{out} > V_{Tn} \]

\[ \Delta V_{out} = -V_{DD} \left( \frac{C_a}{C_a + C_L} \right) \]
Charge Sharing - Solutions

(a) Static bleeder

(b) Precharge of internal nodes

Aside: Dynamic Latch

IN  DYN  OUT
INV1  C_{DYN}
Charge Sharing

\[ A, B = 0 \]
\[ DYN \text{ precharged} \]
Charge sharing if SEL toggles

Aside: Noise in ICs

- **Sources of noise**
  - Coupling
    - Device coupling
    - Capacitive coupling between wires
    - Inductive coupling
  - Supply line bounce
  - Charge Injection
    - From substrate
    - \( \alpha \)-particles
- **Robustness of a circuit**
  - Noise margins
  - Sensitivity to noise
Clock Feedthrough

Miller and Back-gate Coupling
Capacitive Coupling

Dynamic node: Static node:

(a) (b)
Capacitive Coupling

Lateral coupling:

![Capacitive Coupling Diagram](image)

Shielding

![Shielding Diagram](image)

Minority Charge Injection

![Minority Charge Injection Diagram](image)
Supply Noise

![Supply Noise Diagram]

Cascading Dynamic Gates

![Cascading Dynamic Gates Diagram]

Only 0→1 Transitions allowed at inputs!
Cascading Dynamic Logic

![Cascading Dynamic Logic Diagram]

DOMINO LOGIC

![Domino Logic Diagram]

Krambeck et al, JSSC 6/82
Logical Effort

Inverter pair:

\[ \text{Skewed inverter pair:} \]

Logical Effort

\[ LE = \]
Domino Logic - Characteristics

- Only non-inverting logic
- Very fast - Only 1->0 transitions at input of inverter
  move $V_M$ upwards by increasing PMOS
- Adding level restorer reduces leakage and charge redistribution problems
- Optimize inverter for fan-out

Designing with Domino Logic

Can be eliminated!

Inputs = 0 during precharge
Logical Effort

\[ \text{LE} = \phi \text{Out} \]

Delayed Precharge

(a) Inputs to other domino gates
(b) Stage 1 inputs, delayed precharge removed

\[ \phi \text{OUT1} \]
\[ \text{M}_{\phi 1} \]
\[ \text{M}_{\phi 2} \]
\[ \text{INV} \]
\[ \text{DYN} \]
\[ \text{OUT1} \]
\[ \text{OUT2} \]
\[ \text{M}_{\phi 2} \]
\[ \text{M}_{\phi 1} \]

53

54
IBM’s 1GHz Processor

Silberman et al,
ISSCC’98
JSSC 11/98

Domino Properties

- Logic evaluation propagates as falling dominoes
- Evaluation period determines the logic depth
- The nodes must be precharged during the precharge period (can limit the minimum size of PMOS)
- Inputs must be stable (or have only one rising transition) during the evaluation
- Gates are ratioless
- Restorer is ratioed
- All the gates are non-inverting
- Only one transition to be optimized
Logic Design Problem

- How to design an XOR/MUX without a complementary signal available? We need it in datapaths!
- If the logic is followed by a flip-flop, or a latch with a hard edge, can use complementary or pass-transistor logic
- Domino logic is used with latches, and a new domino stage may follow the XOR
- Solutions:
  - Use dual-rail domino (dynamic CVSL)
  - Violate some of domino rules (but still design a reliable circuit)
  - Force a hard edge

Sum Implementation (1)
Sum Implementation (2)

Sum Implementation (3): Strobing

[Anders et al, ISSCC'02]

[Park, VLSI'00]
Differential (Dual Rail) Domino

Dynamic CVSL (Clock CVSL) - Heller et al, ISSCC’84