Arithmetic Building Blocks

- Datapath elements
- Adder design
  - Static adder
  - Dynamic adder
- Multiplier design
  - Array multipliers
- Shifters, Parity circuits

A Generic Digital Processor

MEMORY

CONTROL

DATAPATH

Input-Output
Building Blocks for Digital Architectures

- **Arithmetic unit**
  - Bit-sliced datapath (adder, multiplier, shifter, comparator, etc.)
- **Memory**
  - RAM, ROM, Buffers, Shift registers
- **Control**
  - Finite state machine (PLA, random logic.)
  - Counters
- **Interconnect**
  - Switches
  - Arbiters
  - Bus

Bit-Sliced Design

Tile identical processing elements
Single-Bit Addition

Half Adder

\[ S = A \oplus B \]
\[ C_{out} = A \cdot B \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( C_{out} )</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</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>

Full Adder

\[ S = A \oplus B \oplus C \]
\[ C_{out} = MAJ(A, B, C) \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>( C_{out} )</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Full-Adder

\( C_{in} \leftarrow \uparrow \quad \text{Sum} \leftarrow \downarrow \quad \text{Cout} \quad \text{Cin} \quad \text{A} \quad \text{B} \quad C_{i} \quad S \quad C_{o} \quad \text{Carry status} \)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>( C_{i} )</th>
<th>S</th>
<th>( C_{o} )</th>
<th>Carry status</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>delete</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>delete</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>propagate</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>propagate</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>propagate</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>propagate</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>generate</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>generate</td>
</tr>
</tbody>
</table>
The Binary Adder

\[ \text{Sum} = A \oplus B \oplus C \]
\[ = ABC_i + ABC_i + ABC_i + ABC_i \]
\[ C_o = AB + BC_i + AC_i \]

Sum and Carry as a functions of P, G

Define 3 new variable which ONLY depend on A, B

Generate \((G) = AB\)

Propagate \((P) = A+B\)

\[ C_o(G,P) = G + PC_i \]
\[ S(G,P) = P \oplus C_i \]
The Ripple-Carry Adder

Worst case delay linear with the number of bits
\[ t_d = O(N) \]

\[ t_d = (N-1)t_{\text{carry}} + t_{\text{sum}} \]

Goal: Make the fastest possible carry path circuit

Complementary Static CMOS Full Adder

Note:
1) \( S = ABC_i + \overline{C_i}(A + B + C_i) \)
2) Placement of \( C_i \)

O(N) delay

28 Transistors
**Inversion Property**

Inverting all inputs results in inverted outputs

\[
\bar{S}(A, B, C_i) = S(\bar{A}, \bar{B}, \bar{C_i}) \\
\bar{C}_o(A, B, C_i) = C_o(\bar{A}, \bar{B}, \bar{C_i})
\]

**Minimize Critical Path by Reducing Inverting Stages**

Need two different types of cells, FA': no inverter in carry path
A better structure: the Mirror Adder

The Mirror Adder

- Symmetrical NMOS and PMOS chains
  - Identical rising and falling transitions if the NMOS and PMOS devices are properly sized.
  - Maximum of two series transistors in the carry-generation circuitry.
- Critical issue: minimization of the capacitance at \( C_o \).
  - Reduction of the diffusion capacitances important.
  - The capacitance at \( C_o \) composed of four diffusion capacitances, two internal gate capacitances, and six gate capacitances in the connecting adder cell.
- Transistors connected to \( C_i \) placed closest to output.
- Only the transistors in carry stage have to be optimized for speed. All transistors in the sum stage can be minimal size.
NP-CMOS Adder

17 transistors, ignoring extra inverters for inputs and outputs

Manchester Carry Chain

\[
\begin{align*}
\text{V}_{\text{DD}} & \\
\emptyset & \quad P_0 \\ & \quad P_1 \\ & \quad P_2 \\ & \quad P_3 \\ & \quad P_4 \\ & \quad C_{0,4} \\
C_{1,0} & \quad G_0 \\ & \quad G_1 \\ & \quad G_2 \\ & \quad G_3 \\ & \quad G_4 \\
\emptyset & \\
\end{align*}
\]

- Only nMOS transmission gates used. Why?
- Delay of long series of pass gates: add buffers
Carry-Bypass Adder

Idea: If (P0 and P1 and P2 and P3 = 1) then \( C_{0,3} = C_0 \), else "kill" or "generate".

Manchester-Carry Implementation

\[
\begin{align*}
C_{1,0} & \quad P_0 & \quad P_1 & \quad P_2 & \quad P_3 & \quad BP \\
G_0 & \quad G_1 & \quad G_2 & \quad G_3 & \quad C_{0,3} & \quad BP
\end{align*}
\]
Carry-Bypass Adder (cont.)

Design N-bit adder using N/M equal length stages

\[ t_p = t_{\text{setup}} + M t_{\text{carry}} + (N/M-1) t_{\text{bypass}} + M t_{\text{carry}} + t_{\text{sum}}, \text{ i.e. } O(N) \]

Carry Ripple versus Carry Bypass
Carry-Select Adder
Generate carry out for both “0” and “1” incoming carries

- Setup
- "0" Carry Propagation
  - "0" input
  - P, G
- "1" Carry Propagation
  - "1" input
  - Multiplexer
  - Carry Vector
- Sum Generation

4-bit block for bits k, k+1, k+2, k+3

Carry Select Adder: Critical Path

Bit 0-3
- Setup
- "0" Carry
  - "0" input
  - Multiplexer
  - C₀₄₃
- Sum Generation
  - S₀₃

Bit 4-7
- Setup
- "0" Carry
  - "0" input
  - Multiplexer
  - C₀₄₃
- Sum Generation
  - S₁₄₇

Bit 8-11
- Setup
- "0" Carry
  - "0" input
  - Multiplexer
  - C₀₄₃
- Sum Generation
  - S₁₈₁₁

Bit 12-15
- Setup
- "0" Carry
  - "0" input
  - Multiplexer
  - C₀₄₃
- Sum Generation
  - S₁₂₁₅
Carry-Select Adder: Linear Configuration

Linear Carry Select

\[ t_{\text{add}} = t_{\text{setup}} + \frac{N}{M} t_{\text{carry}} + M t_{\text{max}} + t_{\text{sum}} \]
Square Root Carry Select

\[ t_{\text{add}} = t_{\text{setup}} + P \cdot t_{\text{carry}} + (2^{\text{Nmax}} + t_{\text{sum}} \text{ i.e., } O(\sqrt{N}) \]

Adder Delays - Comparison
Carry Look-Ahead - Basic Idea

Delay “independent” of the number of bits

Carry-Lookahead Adders

- High fanin for large N
- Implement as CLA slices, or use 2nd level lookahead generator

16-bit CLA based on 4-bit slices and ripple carry

Faster implementation

CLA generator
Look-Ahead: Topology

[Diagram showing components labeled as VDD, Gnd, Ci, Co, G3, G2, G1, G0, C0,3, P0, P1, P2, P3, and Ci0.]