Digital Integrated Circuits
A Design Perspective

Arithmetic Circuits


Disclaimer: slides adapted for INE5442/EEL7312 by José L. Güntzel from the book´s companion slides made available by the authors.
A Generic Digital Processor

- INPUT-OUTPUT
- MEMORY
- DATAPATH
- CONTROL
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
An Intel Microprocessor

Itanium has 6 integer execution units like this
Bit-Sliced Design

Tile identical processing elements
Bit-Sliced Datapath

From register files / Cache / Bypass

Multiplexers

Shifter
Adder stage 1
Wiring
Adder stage 2
Wiring
Adder stage 3
Sum Select

To register files / Cache

© Digital Integrated Circuits 2nd
Arithmetic Circuits
Itanium Integer Datapath

Fetzer, Orton, ISSCC’02
Adders
A Full-Adder is shown with inputs A, B, and Cin and outputs Cout and Sum. Below is a table outlining the behavior of the Full-Adder:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cᵢ</th>
<th>S</th>
<th>Cₒ</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

\[ S = A \oplus B \oplus C_i \]

\[ = A\overline{B}\overline{C}_i + \overline{A}B\overline{C}_i + \overline{A}\overline{B}C_i + AB\overline{C}_i \]

\[ C_o = AB + BC_i + AC_i \]
Express Sum and Carry as a function of $P, G, D$

Define 3 new variables which ONLY depend on $A, B$

- **Generate** ($G$) = $AB$
- **Propagate** ($P$) = $A \oplus B$
- **Delete** = $A \overline{B}$

$$C_0(G, P) = G + PC_i$$
$$S(G, P) = P \oplus C_i$$

Can also derive expressions for $S$ and $C_0$ based on $D$ and $P$

Note that we will be sometimes using an alternate definition for

**Propagate** ($P$) = $A + B$
The Ripple-Carry Adder

Worst case delay linear with the number of bits

\[ t_d = O(N) \]

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

Goal: Make the fastest possible carry path circuit
Complimentary Static CMOS Full Adder

28 Transistors
Inversion Property

\[
\bar{S}(A, B, C_i) = S(\overline{A}, \overline{B}, \overline{C_i})
\]

\[
\overline{C_o}(A, B, C_i) = C_o(\overline{A}, \overline{B}, \overline{C_i})
\]
Minimize Critical Path by Reducing Inverting Stages

Exploit Inversion Property
A Better Structure: The Mirror Adder

24 transistors
Mirror Adder

Stick Diagram

\[ V_{DD} \]

\[ \text{GND} \]
The Mirror Adder

• The NMOS and PMOS chains are completely symmetrical. A maximum of two series transistors can be observed in the carry-generation circuitry.

• When laying out the cell, the most critical issue is the minimization of the capacitance at node $C_o$. The reduction of the diffusion capacitances is particularly important.

• The capacitance at node $C_o$ is composed of four diffusion capacitances, two internal gate capacitances, and six gate capacitances in the connecting adder cell.

• The transistors connected to $C_i$ are placed closest to the output.

• Only the transistors in the carry stage have to be optimized for optimal speed. All transistors in the sum stage can be minimal size.
Transmission Gate Full Adder

Setup

Sum Generation

Carry Generation
Manchester Carry Chain

\[ C_i \quad P_i \quad G_i \quad D_i \quad V_{DD} \quad C_o \]

\[ P_i \quad G_i \quad V_{DD} \quad C_i \quad C_o \]

© Digital Integrated Circuits 2nd

Slide 20

Arithmetic Circuits
Manchester Carry Chain
Manchester Carry Chain

Stick Diagram

Propagate/Generate Row

Inverter/Sum Row

© Digital Integrated Circuits 2nd

Slide 22

Arithmetic Circuits
Carry-Bypass Adder

Also called Carry-Skip

Idea: If \((P_0 \text{ and } P_1 \text{ and } P_2 \text{ and } P_3 = 1)\) then \(C_{o3} = C_0\), else “kill” or “generate”.

© Digital Integrated Circuits 2nd
Arithmetic Circuits
Carry-Bypass Adder (cont.)

\[ t_{adder} = t_{setup} + M t_{carry} + (N/M-1) t_{bypass} + (M-1) t_{carry} + t_{sum} \]
Carry Ripple versus Carry Bypass

\[ t_p \]

\[ N \]

ripple adder

bypass adder

4.8