Manchester Carry Chain
Manchester Carry Chain
Manchester Carry - Stick Diagram
Carry-Bypass Adder

Idea: If \((P_0 \text{ and } P_1 \text{ and } P_2 \text{ and } P_3 = 1)\) then \(C_{0,3} = C_0\), else “kill” or “generate”.

\[BP = P_0P_1P_2P_3\]
Carry-Bypass Adder

\[ t_{\text{adder}} = t_{\text{setup}} + M t_{\text{carry}} + \left(\frac{N}{M}-1\right) t_{\text{bypass}} + (M-1) t_{\text{carry}} + t_{\text{sum}} \]
Carry Ripple vs Carry Bypass
Carry-Select Adder

Setup

"0" → "0" Carry Propagation

"1" → "1" Carry Propagation

Multiplexer

Sum Generation

$C_{o,k-1}$ → $C_{o,k+3}$
Carry Select Adder: Critical Path
Linear Carry Select

\[ t_{\text{add}} = t_{\text{setup}} + \left( \frac{N}{M} \right) t_{\text{carry}} + M t_{\text{mux}} + t_{\text{sum}} \]
Square Root Carry Select
Adder Delays - Comparison

$\tau_p$ (in unit delays)

$N$

Ripple adder
Linear select
Square root select
LookAhead - Basic Idea

\[ C_{i,k} = f(A_k, B_k, C_{i,k-1}) = G_k + P_k C_{i,k-1} \]
Look-Ahead: Topology

Expanding Lookahead equations:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}C_{o,k-2})$$

All the way:

$$C_{o,k} = G_k + P_k(G_{k-1} + P_{k-1}(... + P_1(G_0 + P_0C_{i,0})))$$
Logarithmic Look-Ahead Adder

$$t_p \sim N$$

$$t_p \sim \log_2(N)$$
Carry Lookahead Trees

\[
C_{o,0} = G_0 + P_0C_{i,0}
\]

\[
C_{o,1} = G_1 + P_1G_0 + P_1P_0C_{i,0}
\]

\[
C_{o,2} = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_{i,0}
\]

\[
= (G_2 + P_2G_1) + (P_2P_1)(G_0 + P_0C_{i,0}) = G_{2:1} + P_{2:1}C_{o,0}
\]

Can continue building the tree hierarchically.
Tree Adders

16-bit radix-4 Kogge-Stone Tree
Sparse Trees

16-bit radix-2 sparse tree with sparseness of 2
Tree Adders

Brent-Kung Tree
Example: Domino Adder

Propagate

\[ P_i = a_i + b_i \]

Generate

\[ G_i = a_i b_i \]
Example: Domino Adder
Example: Domino Sum
Multipliers
The Binary Multiplication

\[ Z = \hat{X} \times Y = \sum_{k=0}^{M-N-1} Z_k 2^k \]

\[ = \left( \sum_{i=0}^{M-1} X_i 2^i \right) \left( \sum_{j=0}^{N-1} Y_j 2^j \right) \]

\[ = \sum_{i=0}^{M-1} \left( \sum_{j=0}^{N-1} X_i Y_j 2^{i-j} \right) \]

with

\[ X = \sum_{i=0}^{M-1} X_i 2^i \]

\[ Y = \sum_{j=0}^{N-1} Y_j 2^j \]
The Binary Multiplication

\[
\begin{array}{c}
1 & 0 & 1 & 0 & 1 & 0 \\
\times & 1 & 0 & 1 & 1 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
\hline
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
\]

Multiplicand
Multiplier
Partial products
Result
The Array Multiplier
The MxN Array Multiplier
— Critical Path

\[ t_{multi} \approx [(M-1)+(N-2)] t_{carry} + (N-1) t_{sum} + (N-1) t_{and} \]
Carry-Save Multiplier

\[ t_{\text{mult}} = (N-1)t_{\text{carry}} + (N-1)t_{\text{and}} + t_{\text{merge}} \]
Multiplier Floorplan

X and Y signals are broadcasted through the complete array.
(→→→)
Wallace-Tree Multiplier

Partial products

First stage

Second stage

Final adder
Wallace-Tree Multiplier

Partial products

First stage

Second stage

Final adder

\[ z_7 \quad z_6 \quad z_5 \quad z_4 \quad z_3 \quad z_2 \quad z_1 \quad z_0 \]
Wallace-Tree Multiplier
Multipliers — Summary

- Optimization Goals Different Vs Binary Adder
- Once Again: Identify Critical Path
- Other possible techniques
  - Logarithmic versus Linear (Wallace Tree Mult)
  - Data encoding (Booth)
  - Pipelining

FIRST GLIMPSE AT SYSTEM LEVEL OPTIMIZATION
Shifters
The Binary Shifter
The Barrel Shifter

Area Dominated by Wiring
4x4 barrel shifter

\[ \text{Width}_{\text{barrel}} \sim 2 \, p_m \, M \]
Logarithmic Shifter
0-7 bit Logarithmic Shifter

\[ width_{\log} p_m \left( 2K + \left( 1 + 2 + \ldots + 2^{K-1} \right) \right) = p_m \left( 2^K + 2K - 1 \right) \]
Power and Speed Trade-Offs in Datapath Structures
Constraints on Digital designs

- Latency constrained
- Throughput constrained

Techniques to achieve constraints

- Enable Time
- Targeted Dissipation source
Design-Time Power Reduction Techniques

- Reducing the Supply Voltage
- Using multiple supply voltages
- Module level voltage selection
- Multiple supplies inside a block
- Using multiple device thresholds
- Reducing Switching Capacitance through Transistor Sizing
- Reducing Switching Capacitance through Logic and Architecture Optimizations
- Reducing Switching Activity by resource allocation
- Reducing Glitching through Path Balancing
Run-Time Power Management

- Dynamic Supply Voltage Scaling
- Dynamic Threshold Scaling

Sleep and Standby Mode
Design Concepts

- Use right structure
- Determine critical timing path
- Wiring and number of vias
- Regularity and modularity
- Power and speed trade-offs