EE241 - Spring 2000
Advanced Digital Integrated Circuits

Lecture 16
Adders

Carry-Skip Adder

Idea: If (P0 and P1 and P2 and P3 = 1)
then C03 = C0, else "kill" or "generate".
MacSorley, Proc IRE 1/61
Lehman, Burla, IRE Trans on Comp, 12/61
Carry-Skip Adder

For an N-bit adder with k-bit groups, the critical path delay can be expressed as:

\[ t_d = (k-1)t_{RCA} + \left( \frac{N}{k} - 2 \right) t_{SKIP} + (k-1)t_{RCA} \]

Critical path delay with constant groups:

\[ t_d = 2(k-1)t_{RCA} + \left( \frac{N}{k} - 2 \right) t_{SKIP} \]
Carry-Skip Adder

Variable Group Length

\[ x_1 = x_4 = 4, \quad x_2 = x_7 = 7, \quad x_3 = x_8 = 10, \quad x_5 = x_6 = 11. \]

\[ t_d = c_1 + \sqrt{c_2 N + c_3} \]

Oklobdzija, Barnes, Arith’85

---

Carry-Skip Adder

Variable Block Lengths

\[ n=64 \quad \Delta mT=24 \]

UC Berkeley EE241
Manchester Carry Chain

- Implement P with pass-transistors
- Implement G with pull-up, kill (delete) with pull-down
- Use dynamic logic to reduce the complexity and speed up

\[ \sum_{i=1}^{N} \left( \sum_{j=1}^{i} R_j \right) \]

\[ t_p = 0.69 \left( \sum_{i=1}^{N} C_i \right) \]

Speed (normalized by 0.69RC) vs. Area (in minimum size devices)


Sizing Manchester Carry Chain
Manchester Chain with Carry-Skip

Delay model:

PTL with SA-F/F Implementation

Matsui,
JSSC 12/94
Propagate and Generate Signals

Define 3 new variable which ONLY depend on A, B

Generate \( (G) = AB \)

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

Delete = \( A \cdot B \)

\[
C_o(G, P) = G + P C_i
\]

\[
S(G, P) = P \oplus C_i
\]

Can also derive expressions for \( S \) and \( C_o \) based on \( D \) and \( P \)

Carry Lookahead Adder

Lookahead Adder

Lookahead Equations

Position $i$: $c_i = g_i + p_ic_{i-1}$

Position $i+1$: $c_{i+1} = g_{i+1} + p_{i+1}c_i$

$= g_{i+1} + p_{i+1}(g_i + p_ic_{i-1})$

$= g_{i+1} + p_{i+1}g_i + p_{i+1}p_ic_{i-1}$

Carry exists if:
- generated in stage $i+1$
- generated in stage $i$ and propagated through $i+1$
- propagated through both $i$ and $i+1$

Lookahead Adder

- Unrolling of carry recurrence can be continued
- If unrolled to level $k$, resulting in two-level AND-OR structure
- AND Fan-In = $k + 1$, OR Fan-In = $k + 1$
- $k + 1$ transistors in the MOS stack
- Limits $k$ to 3 - 4
Lookahead Adder

Block Lookahead

Fourth bit carry:
\[ c_{i+4} = g_{i+3} + p_{i+3} g_{i+2} + p_{i+3} p_{i+2} g_{i+1} + p_{i+3} p_{i+2} p_{i+1} g_{i} + p_{i+3} p_{i+2} p_{i+1} p_{i} c_{i-1} \]

Block generate and block propagate:
\[ G_{i,i+3} = g_{i+3} + p_{i+3} g_{i+2} + p_{i+3} p_{i+2} g_{i+1} + p_{i+3} p_{i+2} p_{i+1} g_{i} + p_{i+3} p_{i+2} p_{i+1} p_{i} c_{i-1} \]
\[ P_{i,i+3} = p_{i+3} p_{i+2} p_{i+1} p_{i} \]
\[ c_{i+4} = G_{i,i+3} + P_{i,i+3} c_{i-1} \]
Block Lookahead

Can create groups of groups, or 'super-groups':

\[ G_j^* = G_{j+3} + P_{j+3}G_{j+2} + P_{j+3}P_{j+2}G_{j+1} + P_{j+3}P_{j+2}P_{j+1}G_j \]

\[ P_j^* = P_{j+3}P_{j+2}P_{j+1}P_j \]

Delay is \[ t_d = c_1 \log[N] \]
Lookahead Example

Multiple Output Domino (MODL)

4-bit group generate

4-bit group propagate
64-b Lookahead Example

Lookahead Example

\[ P = A \& B \text{ (propagate)} \]
\[ G = A \oplus B \text{ (generate)} \]
\[ C_0 \text{ (carry-out)} \]
\[ C_1 = G_2 + C_0 P_0 \]
\[ C_2 = G_2 + C_1 P_1 + G_0 P_1 + C_0 P_0 P_1 \]
\[ C_3 = G_2 + C_2 P_2 + G_1 P_2 + G_0 P_1 P_2 + C_0 P_0 P_0 P_2 + P_3 \]
\[ C_4 = G_2 + C_3 P_3 + G_1 P_3 + G_0 P_2 P_3 + G_0 P_1 P_2 P_3 + C_0 P_0 P_1 P_2 P_3 + P_5 \]
\[ C_5 = C_3 P_3 + P_7 \]
\[ G_6 = G_2 + C_5 P_5 + G_7 + G_2 P_5 + G_0 P_5 P_5 \]
\[ \ldots \]
\[ C_{18} = G_{18} + C_1 P_{18} \]
\[ C_{19} = G_{18} + C_0 P_{18} \]
\[ S_n = P_n \oplus C_n \]
Modified CLA

Naini, CICC'92
Conditional Sum Adders

\[ x = 1011101101101101101 \]
\[ y = 0001100110110110110 \]

\[
s^0_i = x_i \oplus y_i \\
s^1_i = x_i \oplus y_i \\
c^0_i = x_i \cdot y_i \\
c^1_i = x_i + y_i
\]

Sklansky,
Trans on Comp
6/60

UC Berkeley EE241
B. Nikolic
TG Conditional Sum

- Serial connection of transmission gates
- Chain length = $1 + \log_2 n$

Rothermel, JSSC 89

2-way MUXes
DPL Conditional Sum

CLA

"Conditional carry select"

Block Conditional Sums

UC Berkeley EE241  B. Nikolic
Carry-Select Adder

Setup

"0" Carry Propagation

"1" Carry Propagation

Multiplexer

Carry Vector

Sum Generation

Carry-Select Adder: Critical Path

Bit 0-3

Setup

"0" Carry

"1" Carry

Multiplexer

C_0,3

C_0,0

Sum Generation

S_0,3

Bit 4-7

Setup

"0" Carry

"1" Carry

Multiplexer

C_0,7

C_0,4

Sum Generation

S_0,7

Bit 8-11

Setup

"0" Carry

"1" Carry

Multiplexer

C_0,11

C_0,8

Sum Generation

S_0,11

Bit 12-15

Setup

"0" Carry

"1" Carry

Multiplexer

C_0,15

C_0,12

Sum Generation

S_0,15
Linear Carry Select

Square Root Carry Select
Logarithmic Lookahead Adders

\[ \text{tp} \sim \log_2(N) \]

Tree Adders

\[ P_G = p_m \cdot p_l \quad \text{m – more significant} \]
\[ G_G = g_m + p_m \cdot g_l \quad \text{l – less significant} \]

Start from the input P, G, and continue up the tree
2-bit groups, then 4-bit groups, ...

\[ (g, p) = (g_m \cdot p_m) \oplus (g_l \cdot p_l) = (g_m + p_m \cdot g_l, p_m \cdot p_l) \]

Kogge, Stone, Trans on Comp,’73
Brent-Kung Adder

\[ t_{\text{add}} \sim \log_2(N) \]

Tree Adders