Bin logic

Add algorithms

Addition

M. J. Flynn

EE 486 Lecture 6: Integer Addition
Can create a group of groups for faster add.

Bypass logic for carries based on bit

Simplified Manchester carry bit

Manchester addresses

Simplified Manchester carry bit

Carry skip

Carry skip logic

Carry addresses

Carry completion

Compute addresses

Compute Manchester carry bit
Rather than selecting by pairs we can select

\[ E_12 \]

\[ N_8 \]

\[ V(G) \]

\[ G \]

\[ +P \]

\[ +P \]

\[ G \]

\[ 3 \]

\[ 1 \]

\[ VP \]

\[ 0 \]

\[ 0 \]

\[ 0 \]

\[ 0 \]

\[ a \] component in a hybrid adder.

\[ = 0 \]

\[ = 1 \]

\[ = 0 \]

\[ = 0 \]

\[ = 0 \]

\[ C^{0,1}_N \]

\[ C^{0,1}_N \]

\[ C^{0,1}_N \]

\[ C^{0,1}_N \]

\[ C^{0,1}_N \]

\[ C^{0,1}_N \]

\[ C^{0,1}_N \]

For different parts the carry outs are

Conditioned sum

Conditioned sum

Conditioned sum

Conditioned sum

Carry select

Carry select
Carry select delay

- Delay consists of digit addition, carry propagation and final sum selection
- Selection is a MUX: \( S_{SE} = S_{E} + S_{A} C_{0} \)

- Delay is \( d = k + 2 \log_{2} n \) bits

Carry propagating adders

- The most widely studied class of adder
- carry look ahead
- carry select
- prefix adders

Group propagate \((P)\) and Group generate \((G')\)

- \( C_{i} = G_{i} + P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} G_{i} + P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} P_{i} P_{i} C_{0} \)

CLA: carry look ahead adders

- Form \( P \) & \( G \) terms for each bit, i
- \( C_{i} = G_{i} + P_{i} \)
- \( C_{i} = G_{i} + P_{i} + P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} + P_{i} G_{i} + P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} P_{i} C_{0} \)
- \( C_{i} = G_{i} + P_{i} + P_{i} G_{i} + P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} P_{i} G_{i} + P_{i} P_{i} P_{i} P_{i} P_{i} P_{i} C_{0} \)

Group propagate \((P)\) and Group generate \((G')\) propagate and end around
the OR tree which ORs them together. Both
but many terms can be shared (prefixed).

Suppose we have $v_{XXX} + 1 = v_{SSSS}$. To find

- $v_{Final}$

- $v_{CLA}$ delay

- $v_{CLA}$ carry

- $v_{CLA}$ and prefix adders

- $v_{CLA}$ delay

- $v_{CLA}$ delay

- $v_{CLA}$ delay

- $v_{CLA}$ delay
This defines the pseudo carry, \( H \).