

VLSI Design

Lecture 18: Shifters and Adders

Shaahin Hessabi

Department of Computer Engineering

Sharif University of Technology

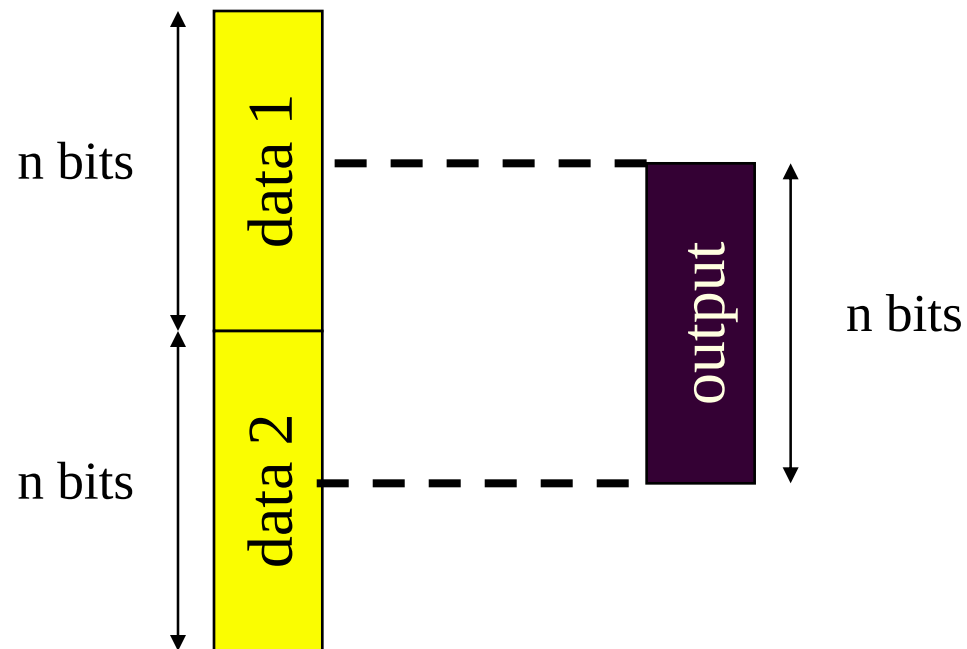
Adapted, with modifications, from lecture notes prepared by the author
(from Prentice Hall PTR)

Combinational shifters

- Useful for arithmetic operations, bit field extraction, etc.
- Latch-based shift register can shift only one bit per clock cycle.
- A multiple-shift shifter requires additional connectivity.

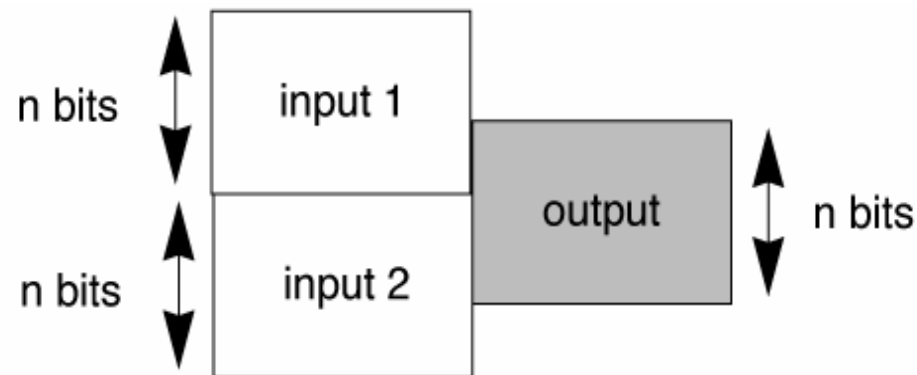
Barrel shifter

- Can perform n -bit shifts in a single cycle.
- Efficient layout.
- Does require transmission gates and long wires.
- Accepts $2n$ data inputs and n control signals, producing n data outputs.



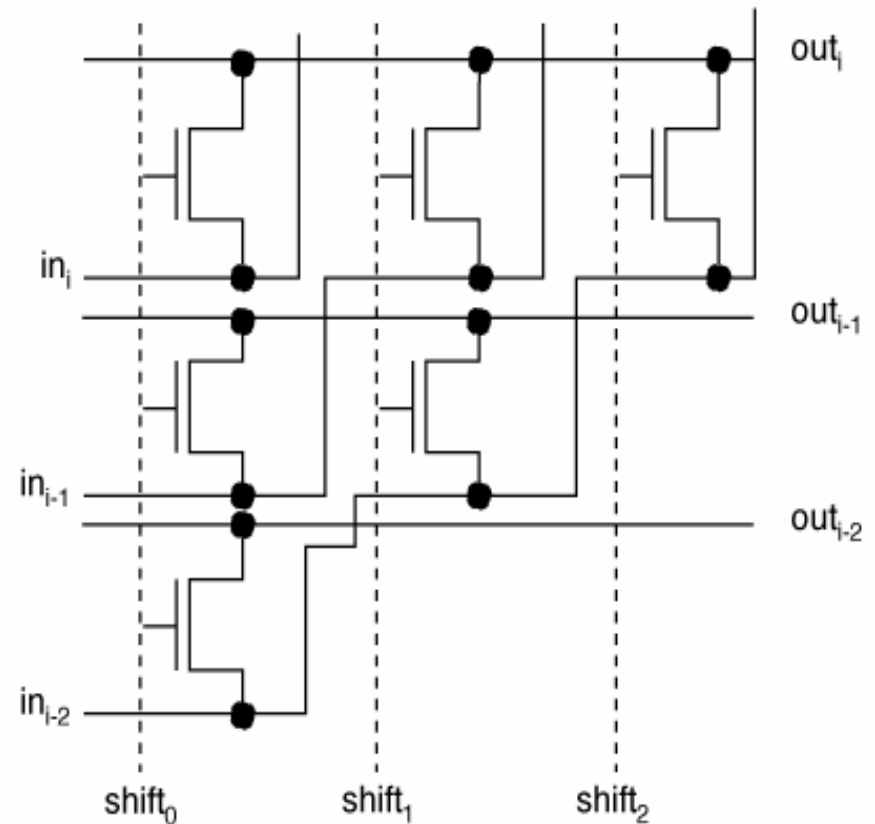
Barrel shifter operation

- Selects arbitrary contiguous n bits out of $2n$ input bits.
- Examples:
 - right shift: data into top, 0 into bottom;
 - left shift: 0 into top, data into bottom;
 - rotate: data into top and bottom.



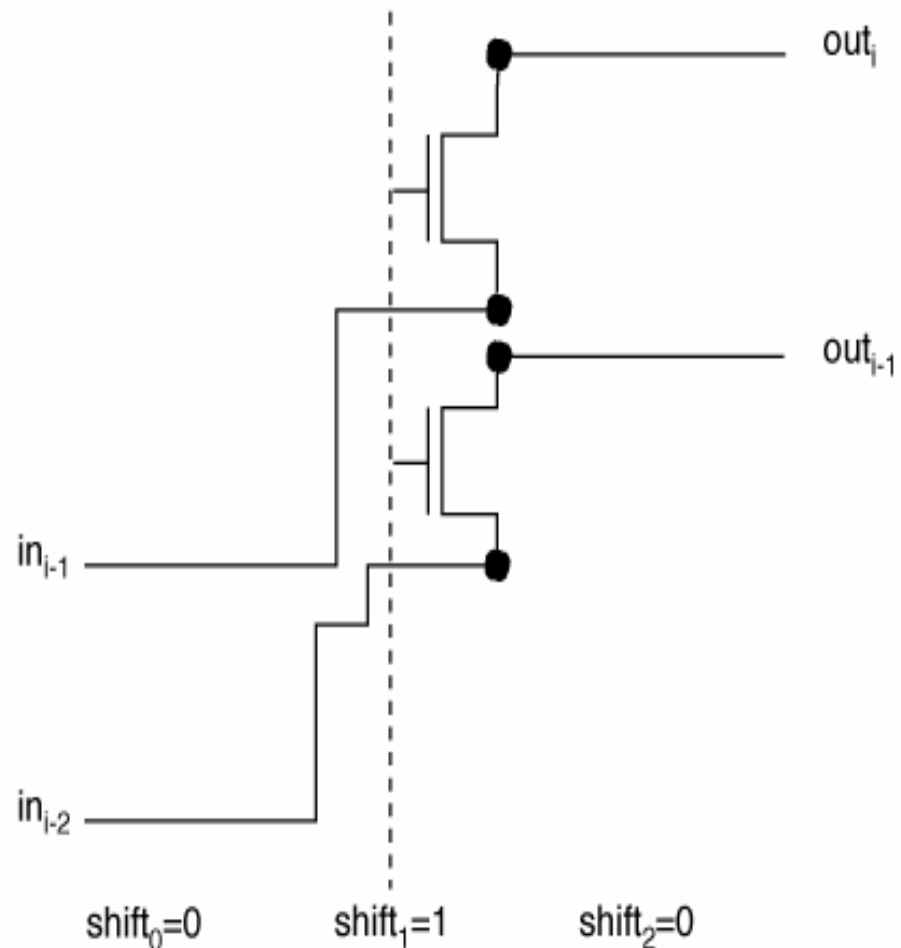
Barrel shifter layout and cell

- Two-dimensional array of $2n$ vertical $\times$ n horizontal cells.
- Input data travels diagonally upward. Output wires travel horizontally.
- Control signals run vertically. Exactly one control signal is set to 1, turning on all transmission gates in that column.



Barrel shifter in action

- Large number of cells, but each one is small.
- Delay is large, considering long wires and transmission gates.



Adders

- Adder delay is dominated by carry chain.
- Carry chain analysis must consider transistor, wiring delay.
- Modern VLSI favors adder designs which have compact carry chains.

Full adder

- Computes one-bit sum, carry:
 - $s_i = a_i \text{ XOR } b_i \text{ XOR } c_i$
 - $c_{i+1} = a_i b_i + a_i c_i + b_i c_i$
- **Ripple-carry adder:** n-bit adder built from full adders.
 - Delay of ripple-carry adder goes through all carry bits.

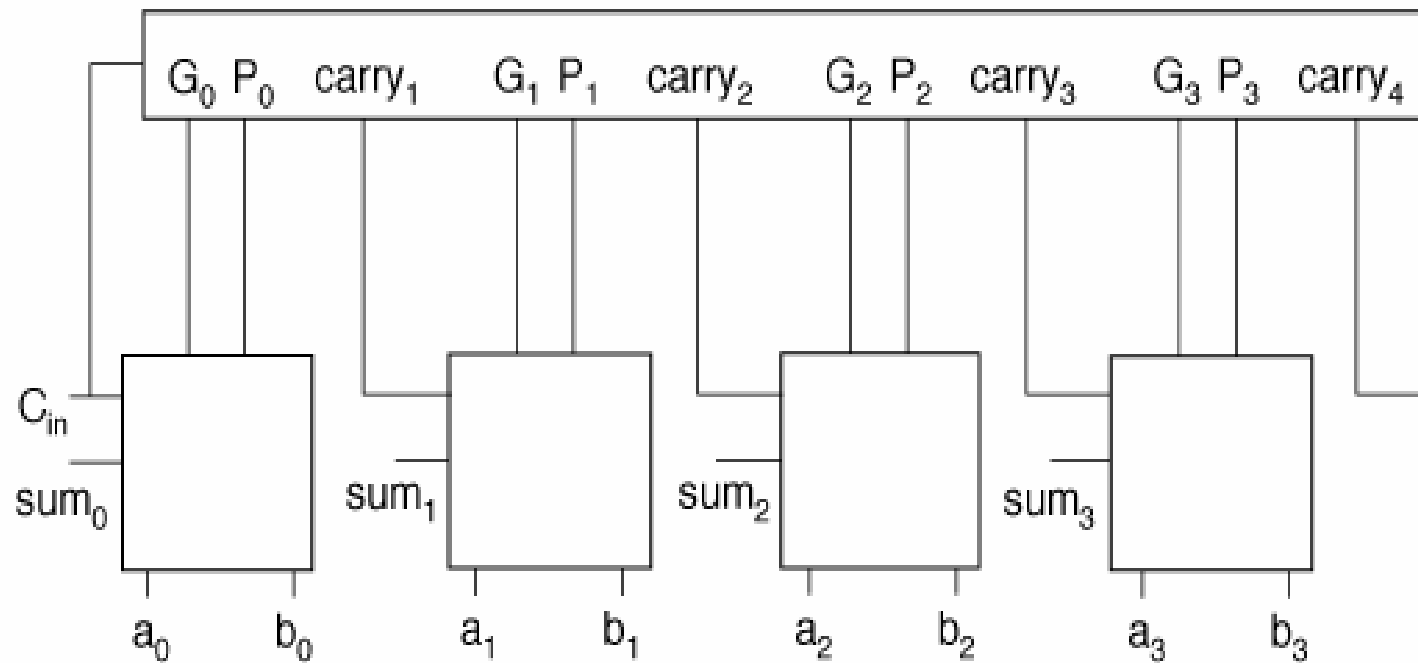
Carry-lookahead adder

- First compute carry propagate, generate:
 - $P_i = a_i + b_i$
 - $G_i = a_i b_i$
- Compute sum and carry from P and G:
 - $s_i = c_i \text{ XOR } P_i \text{ XOR } G_i$
 - $c_{i+1} = G_i + P_i c_i$

Carry-lookahead expansion

- Can recursively expand carry formula:
 - $c_{i+1} = G_i + P_i(G_{i-1} + P_{i-1}c_{i-1})$
 - $c_{i+1} = G_i + P_iG_{i-1} + P_iP_{i-1}(G_{i-2} + P_{i-1}c_{i-2})$
- Expanded formula does not depend on intermediate carries.
- Allows carry for each bit to be computed independently.

Depth-4 carry-lookahead

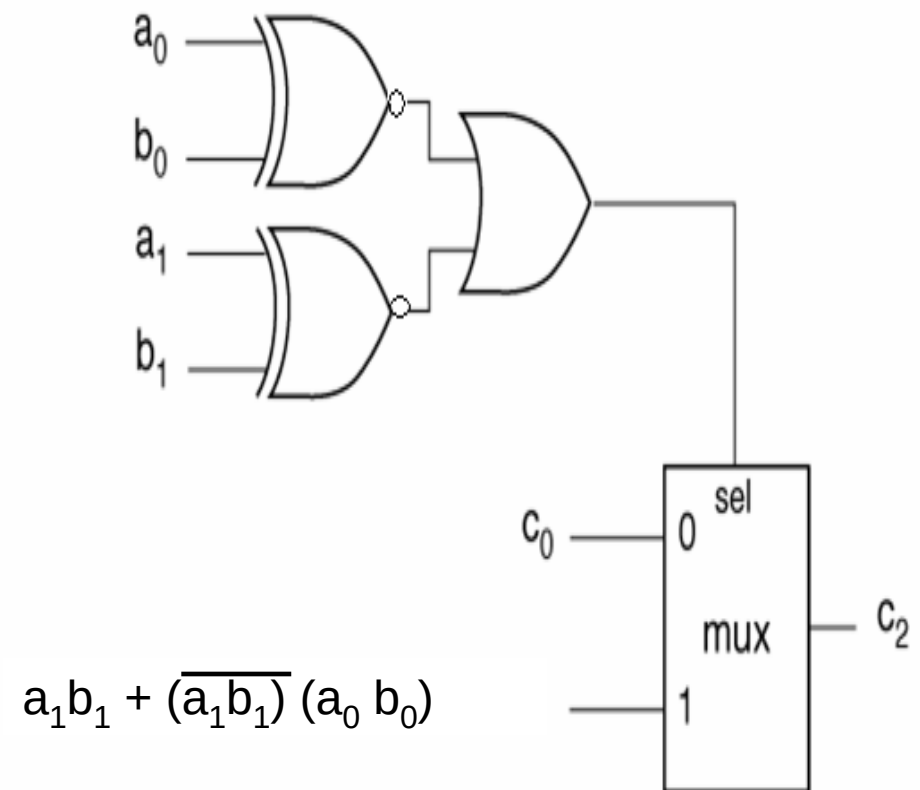


Analysis

- Deepest carry expansion requires gates with large fanin: large, slow.
- Carry-lookahead unit requires complex wiring between adders and lookahead unit; values must be routed back from lookahead unit to adder.
- Layout is even more complex with multiple levels of lookahead.

Carry-skip adder

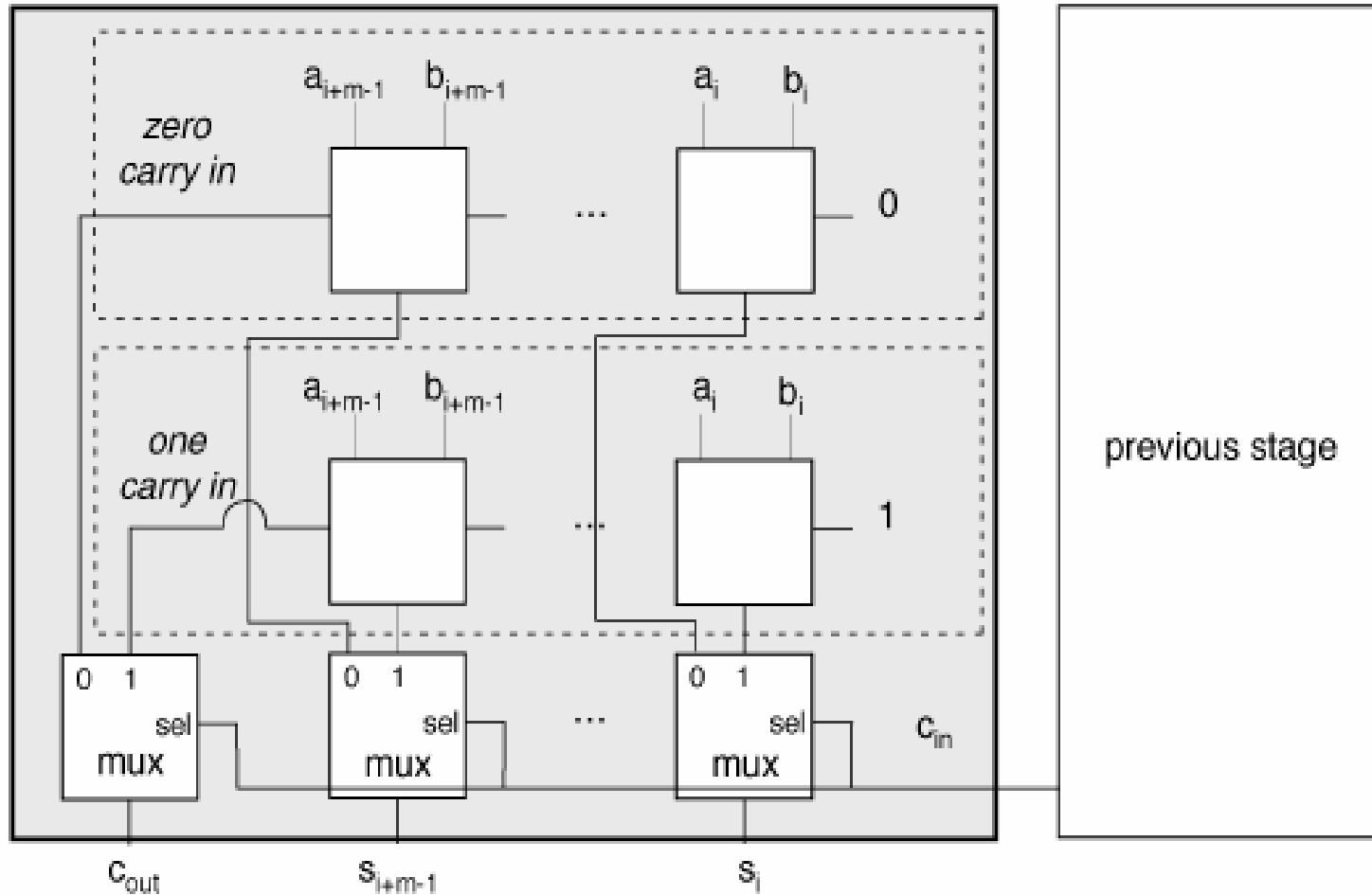
- Looks for cases in which carry-out of a set of bits is identical to carry in.
- Typically organized into m -bit stages.
- If $a_i \neq b_i$ for every bit in stage, then bypass gate sends stage's carry input directly to carry output.
- Figure shows a two-bit carry-skip structure



Carry-select adder

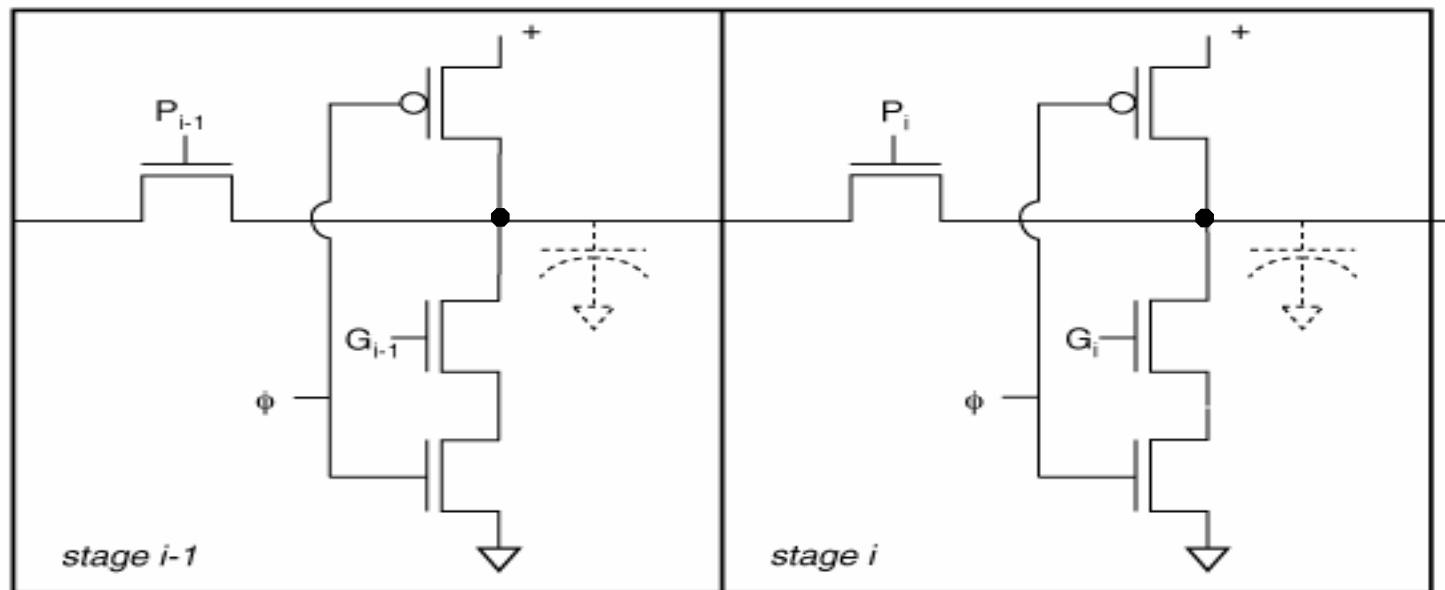
- Computes two results in parallel, each for different carry input assumptions.
- Uses actual carry-in to select correct result.
- Reduces delay to multiplexer.

Carry-select structure



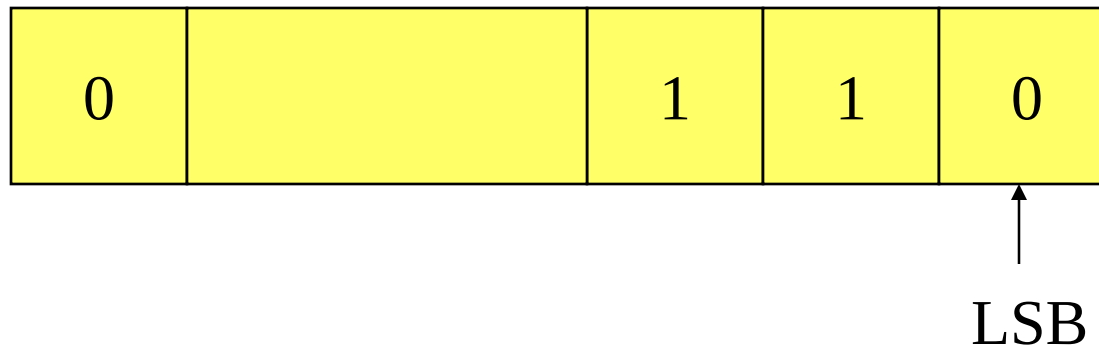
Manchester carry chain

- Precharged carry chain which uses P and G signals.
- *Propagate* signal connects adjacent carry bits.
- *Generate* signal discharges carry bit.
- Worst-case discharge path goes through entire carry chain.



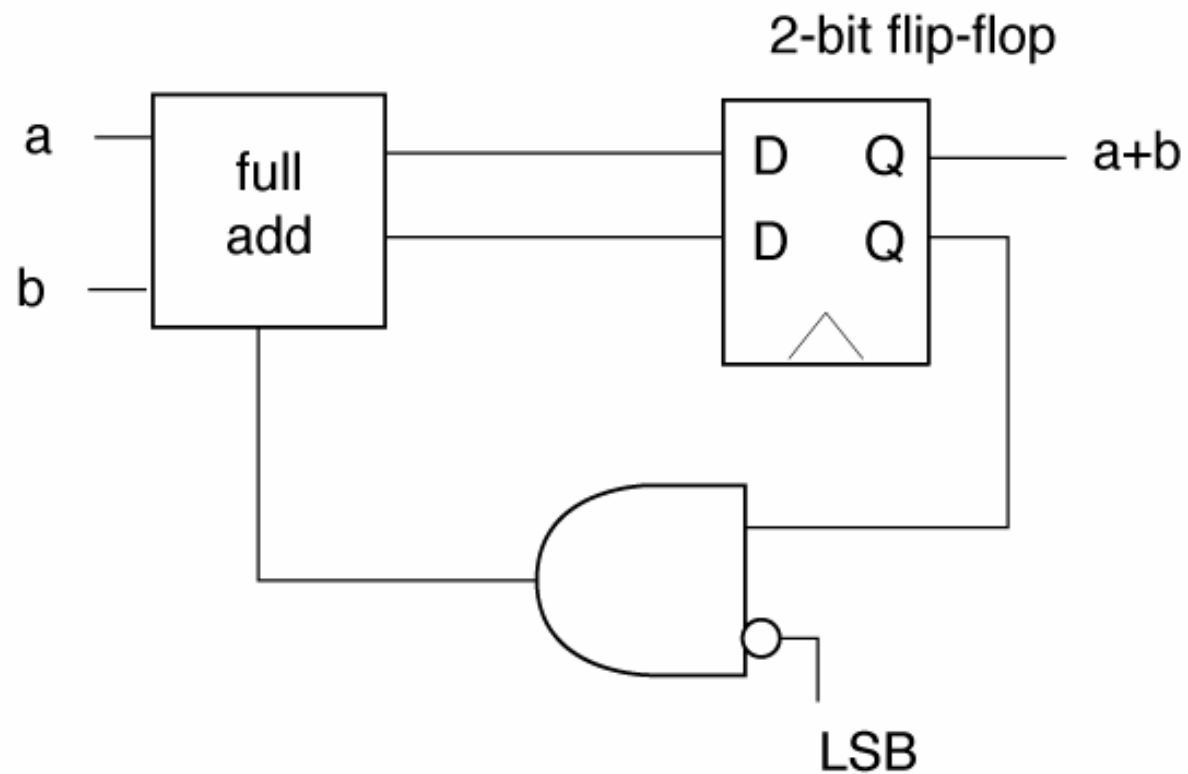
Serial adder

- May be used in signal-processing arithmetic where fast computation is important but latency is unimportant.
- Data format (LSB first):



Serial adder structure

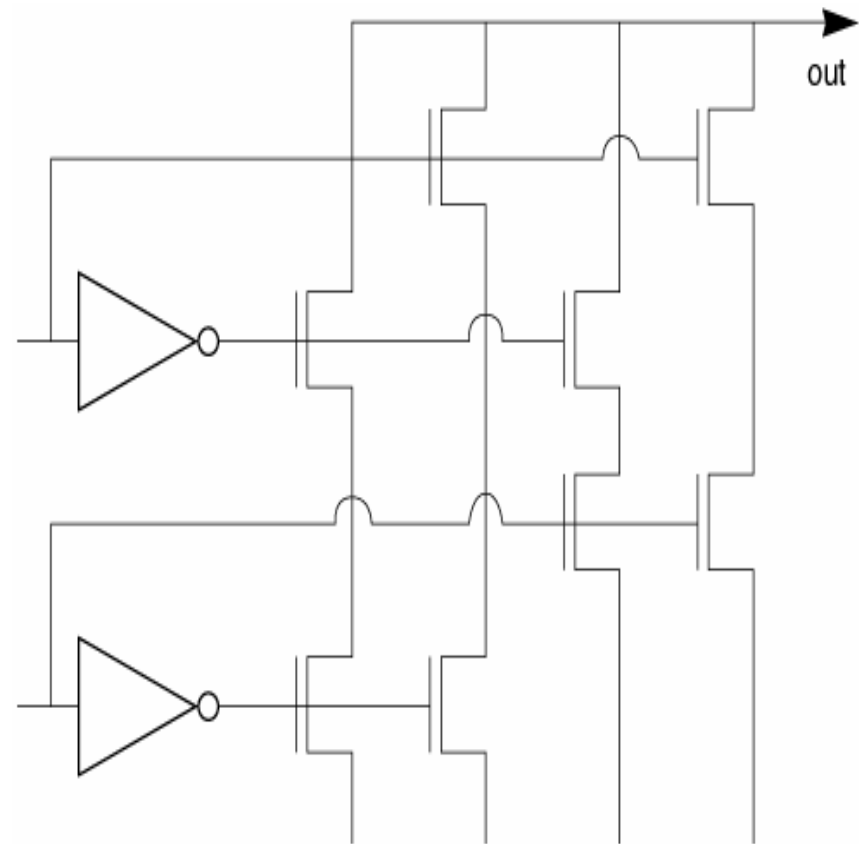
LSB control signal clears the carry shift register:



- ALU computes a variety of logical and arithmetic functions based on **opcode**.
- May offer complete set of functions of two variables or a subset.
- ALU built around adder, since carry chain determines delay.

Function block circuit and ALU

- Function block may be used to compute required intermediate signals for a full-function ALU.
- Requires little area.
- Transmission gates may introduce significant delay.



ALU structure and design

- P and G compute intermediate values from inputs. May not correspond to carry lookahead P and G for non-addition functions.
- Add unit is adder of choice.
- Output unit computes from sum, propagate signal.

