Shifters and Adders

Somayyeh Koohi

Department of Computer Engineering

Sharif University of Technology

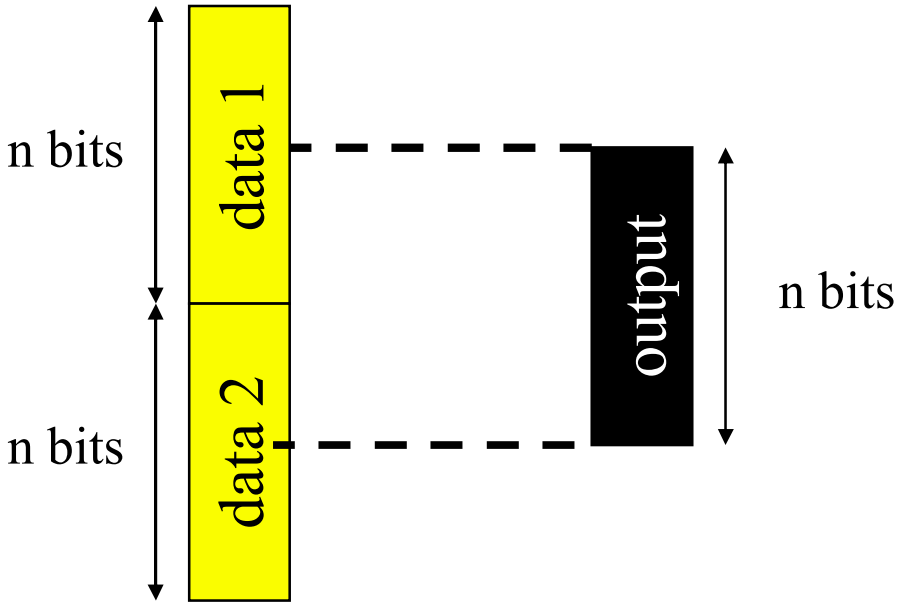
Adapted with modifications from lecture notes prepared by author

Combinational shifters

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

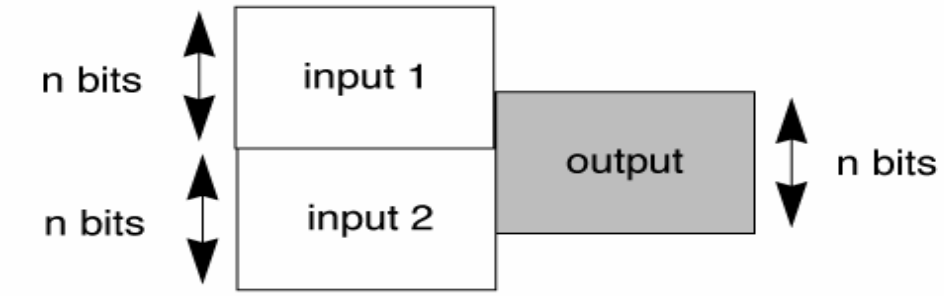
Barrel shifter

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



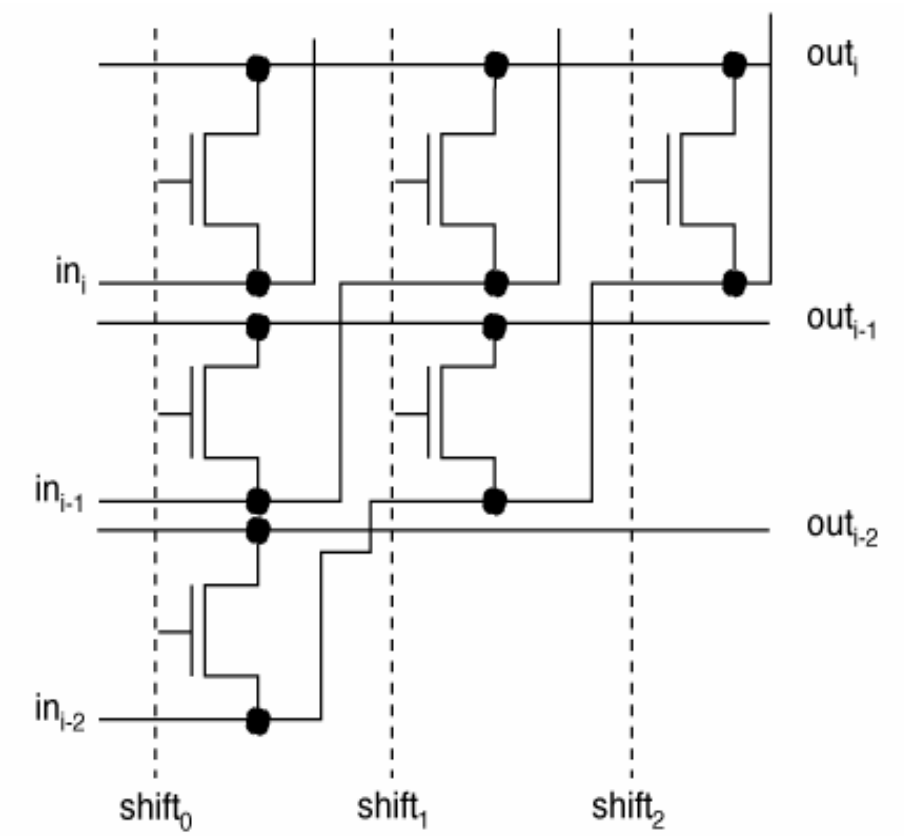
Barrel shifter operation

- Selects arbitrary contiguous *n* bits out of *2n* input buts
- Examples:
  - Right shift: data into top, 0 into bottom
  - Left shift: 0 into top, data into bottom
  - Rotate: data into top and bottom



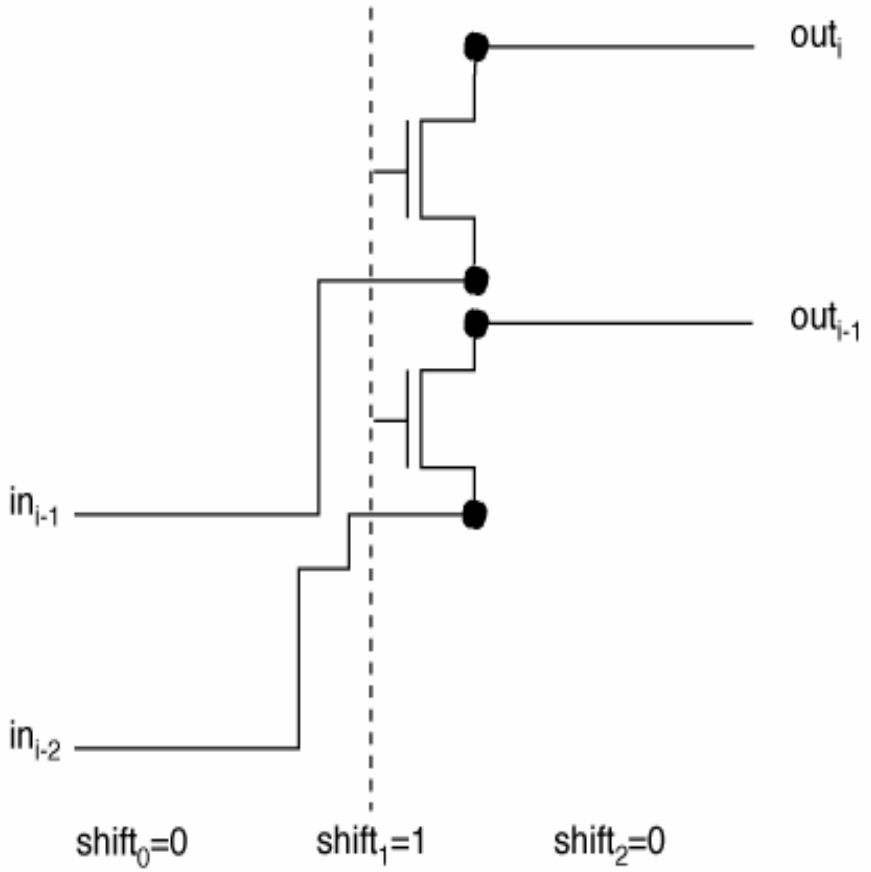
Barrel shifter layout and cell

- Two-dimensional array of *2n* vertical X *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
  - Due to long wires
    - Large delay
  - Each signal traverse only one transmission gate from input to output
    - Small delay due to transmission gates



Adders

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

Full adder

- Computes one-bit sum, carry:
  - $s_i = a_i$ XOR b_i XOR c_i
  - $c_{i+1} = a_i b_i + a_i c_i + b_i c_i$
- Various adder architectures differ from each other in carry generation circuit
  - Ripple carry adder
  - Carry-lookahead adder
  - Carry-select adder
  - Carry skip adder
  - Manchester carry chain
  - Serial adder

Ripple carry adder

- n-bit adder built from cascading 1-bit full adders
  - Delay of ripple-carry adder goes through all carry bits
- Efficient in FPGAs
  - Implement carry chain
    - E.g. : Xilinx XC4000
- The worst in ASICs

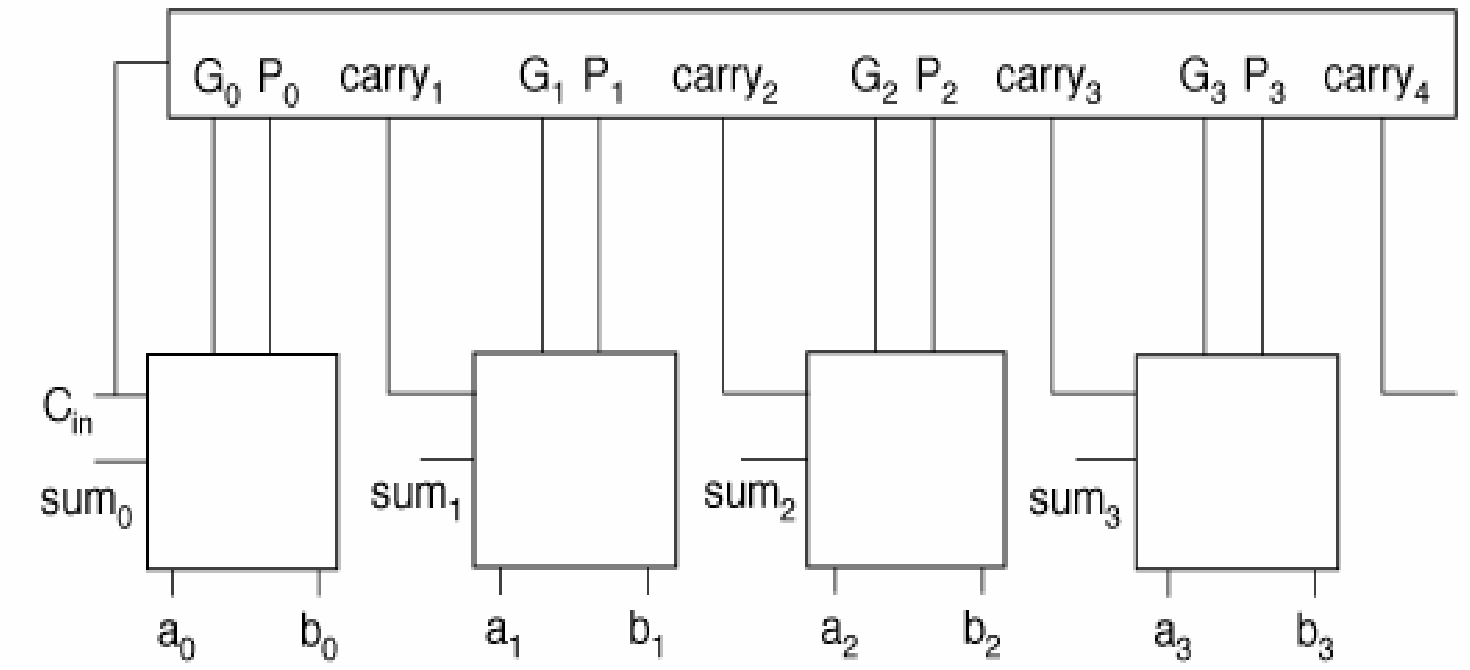
Carry-lookahead adder

- 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:
  - $s_i = c_i$ XOR P_i XOR G_i
  - $c_{i+1} = G_i + P_i c_i$

Carry-lookahead expansion

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

Depth-4 carry-lookahead

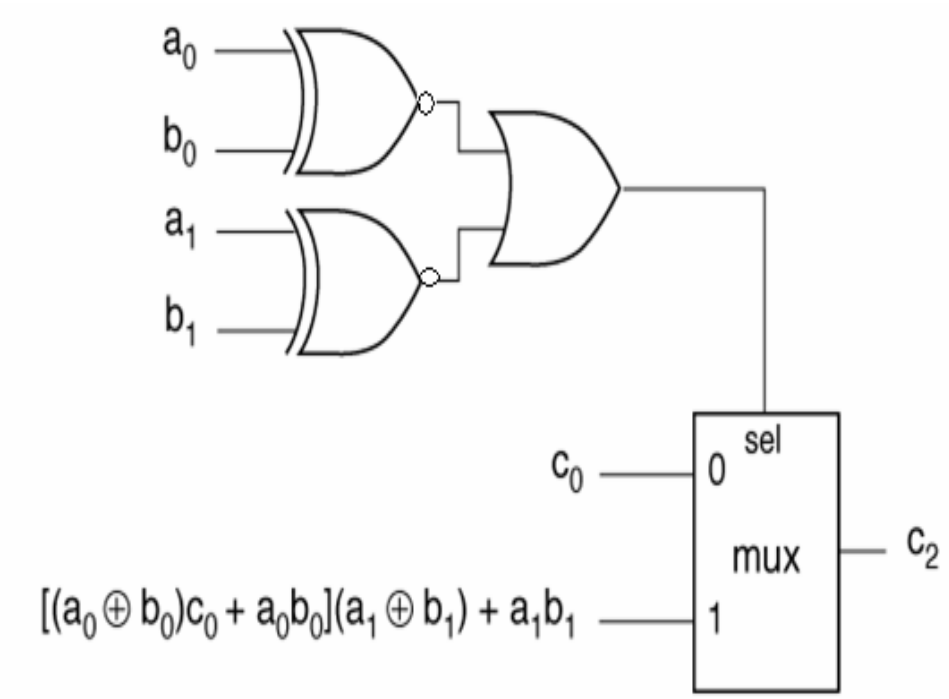


Analysis

- Increased complexity and area, reduced delay for small adder (e.g. 4-bit adder)
- Deepest carry expansion requires gates with large fanin
  - So, slow
  - Limit number of inputs to 4 (depth-4 carry-lookahead) for each carry lookahead adder
    - So, hierarchical structure
- 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
  - Use ripple carry or carry-lookahead structure between different levels

Carry-skip adder

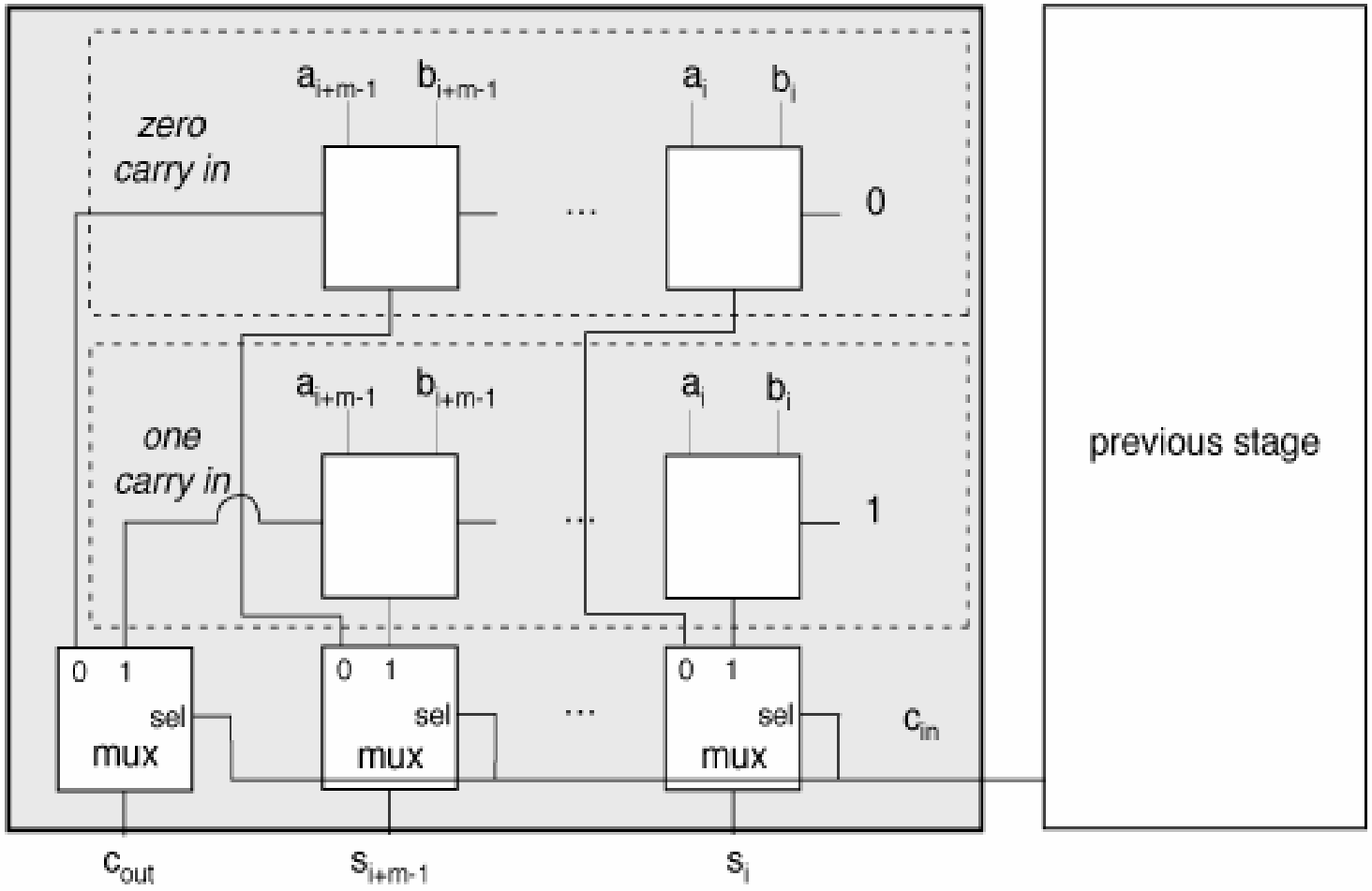
- Looks for cases in which carry-out of a set of bits is identical to carry in
- Typically organized into *m*-bit stages
- If $a_i \neq b_i$ for every bit in stage
  - Bypass gate sends stage's carry input directly to carry output
  - O.W : generate carry
- Figure : two-bit carry-skip structure



Carry-select adder

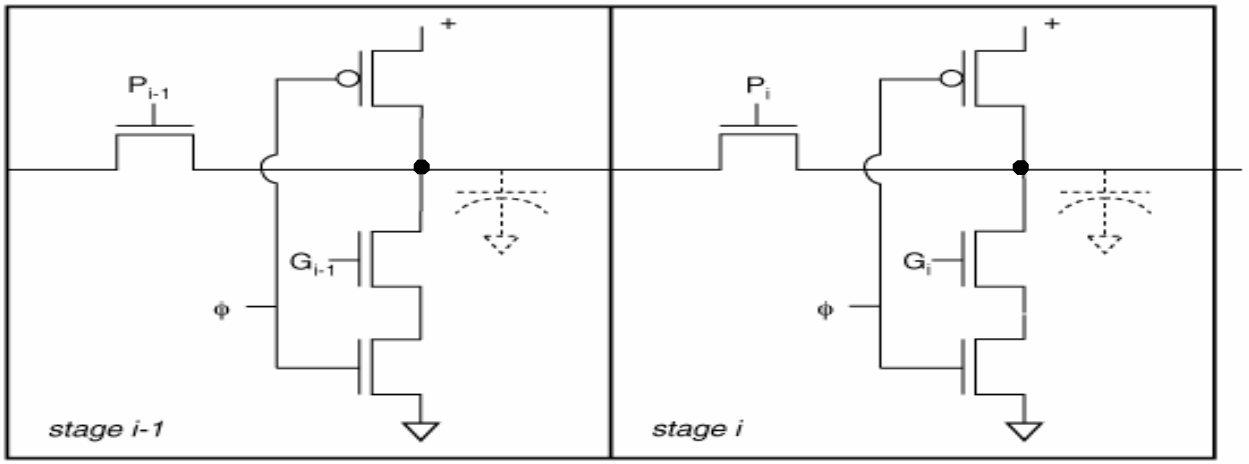
- Computes two results in parallel
  - Each for different carry input assumptions
- Uses actual carry-in to select correct result
- Reduces delay to multiplexer
- Carry-select adder can be divided to m-bit sub adders
  - Sub adder implemented as carry select adder
  - Carry ripple between sub adder
  - 32-bit adder : 8 mux-stage delay

Carry-select structure



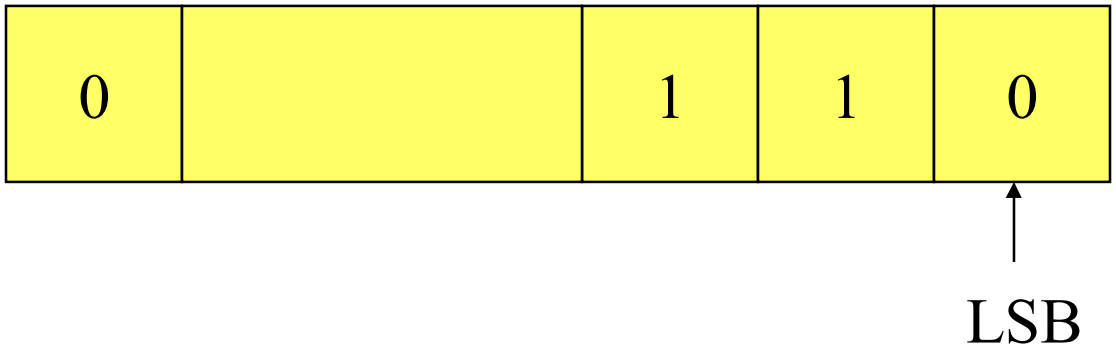
Manchester carry chain

- Precharged carry chain which uses P and G signals
  - Propagate signal connects adjacent carry bits
  - Generate signal discharges carry bit
- Storage node holds the complement of the carry
- Worst-case discharge path goes through entire carry chain
- Widest transistor should be at the least-significant bit stage
  - They see the largest load



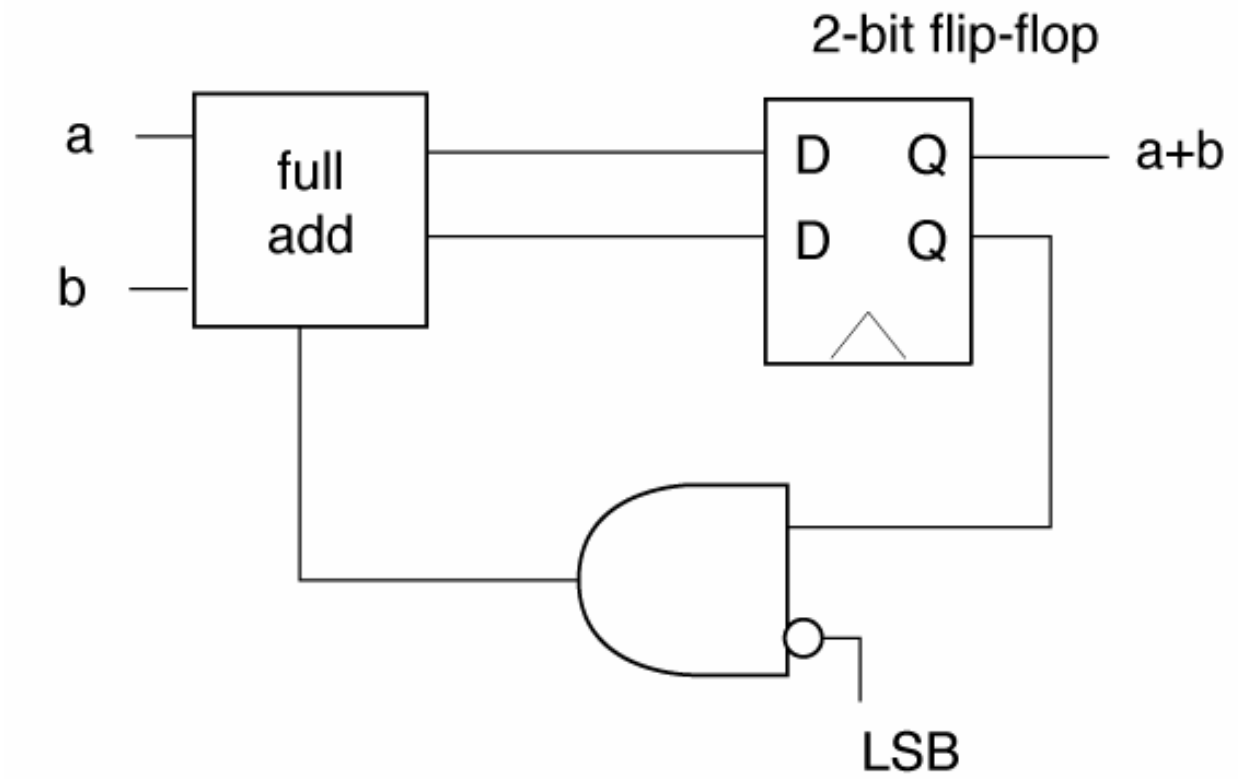
Serial adder

- May be used in signal-processing arithmetic
  - Fast computation is important
- Requires many clock cycle to add two n-bit numbers
  - But with very short cycle time
- Simple and small
- Data format (LSB first):



Serial adder structure

LSB control signal clears the carry shift register:

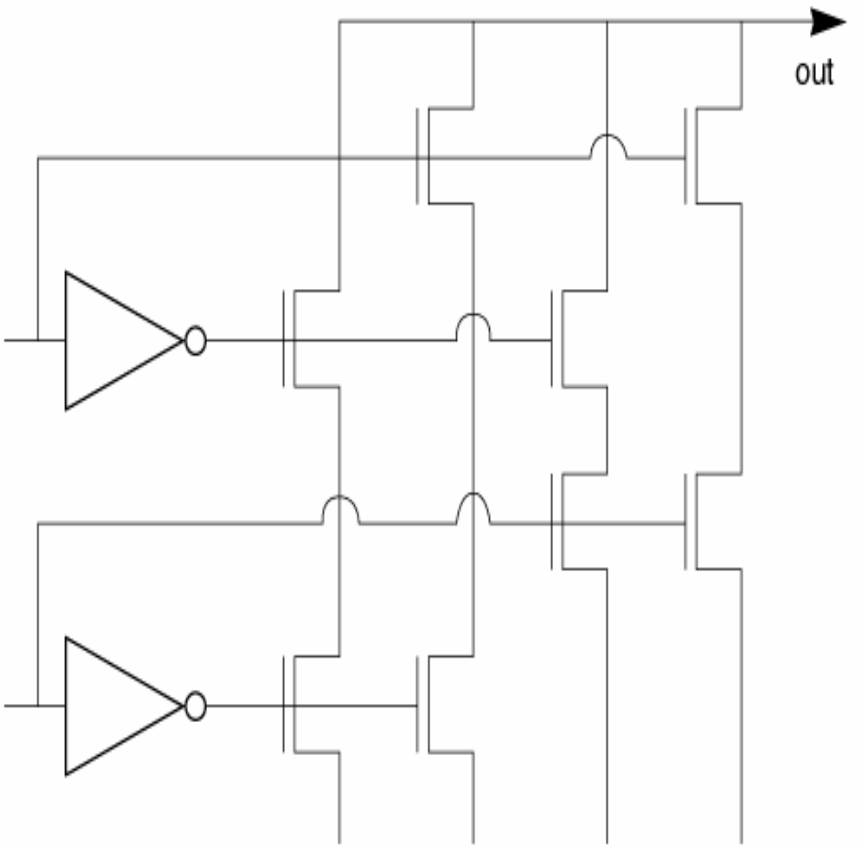


ALUs

- ALU computes a variety of (bit-wise) logical and arithmetic functions based on opcode
  - Opcode together with carry-in determine the function
- 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

- Takes 2 data inputs and their complement along with 4 control signals
  - Can compute all 16 possible functions of the 2 data inputs
- Function block may be used to compute required intermediate signals for a full-function ALU
- Requires little area



ALU structure and design

- 3 function blocks
  - Require a total of 12 opcode
- 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