CSED601
Dependable Computing
Lecture 9

Jong Kim

Dept. of CSE

POSTECH

Review of Previous Lecture

- Code types
  - Cyclic code, Cyclic Redundancy Check (CRC)
  - Arithmetic Code
    - AN code, Residue code, Inverse residue code, RNS
  - Berger code
  - Horizontal and Vertical parity code
  - Hamming Error Correcting code
  - Code selection issues
  - Error model

Time Redundancy

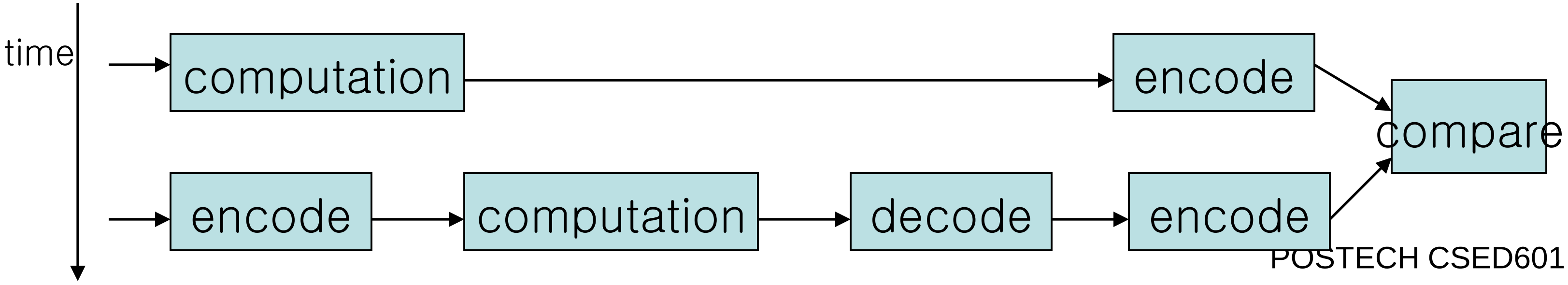
- Concept
  - The penalty paid is time.
  - Attempt to reduce the amount of extra hardware at the expense of using additional time
  - Time redundancy can detect a permanent fault with a minimum extra hardware
  - The selection of a particular type of redundancy is very dependant upon the application
- Methods
  - Retry
  - Alternating logic
  - Re-computing with shifted operands (RESO)
  - Re-computing with swapped operands (RWSO)
  - Re-computing with duplication with comparison (REDWC)

Retry

- Concept
  - Used to detect transient fault
  - Repeat the same computation
- Methods
  - Method 1:
    - The repetition of computations in ways that allow faults to be detected.
    - Perform the same computation two or more times and compare the results to determine if a discrepancy exists.
  - Method 2:
    - The detection of fault is done by using other mechanism
    - Time redundancy is employed to distinguish between the permanent and the transient fault
  - Problem: Assure that the system has the same data to manipulate whenever it performs a computation redundantly.
  - Combine time and information redundancy
    - Use 3N and 5N codes for repetitive tries

Alternating Logic

- Concept
  - Compute with original data and compute with complemented data
  - Applied to the transmission of digital data over wire media
  - The concept can be applied to to general combinational logic circuit.
- Alternating logic
  - Using the self-duality
    - If the circuit possesses the property of self-duality, then f(x) = f'(x')
    - Full-adder is a self-dual circuit.
  - Advantage
    - Any circuit can be transformed into a self-dual circuit by adding one more variable.

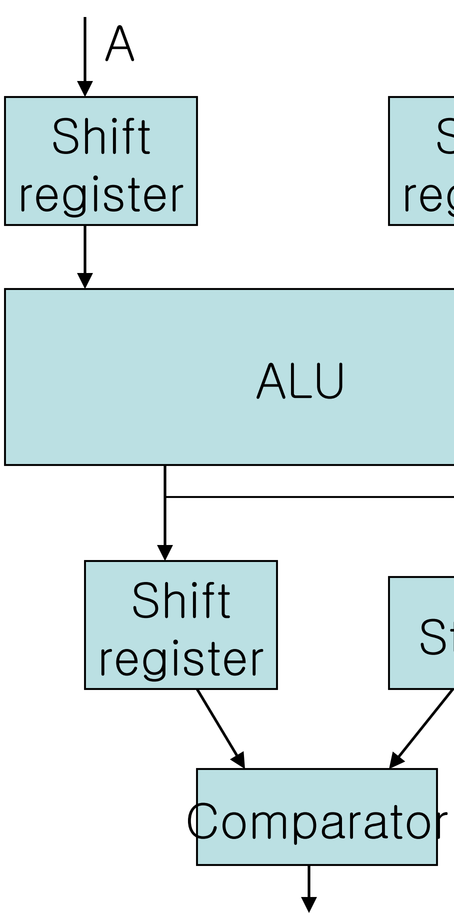


Re-computing with shifted operands (RESO)

- Concept
  - Developed as a method to provide concurrent error detection in ALUs
  - Encoding function: left-shift operation
  - Decoding function: right-shift operation
  - Assume bit-sliced hardware
  - Eg.: Ripple carry adder with additional bit slice
- Problem
  - Additional hardware is required
  - Lack of coverage provided in the shifters
  - Comparators should be TSC (totally self-checking circuit)

Re-computing with shifted operands (RESO)

- Model



- When bit i is faulty,

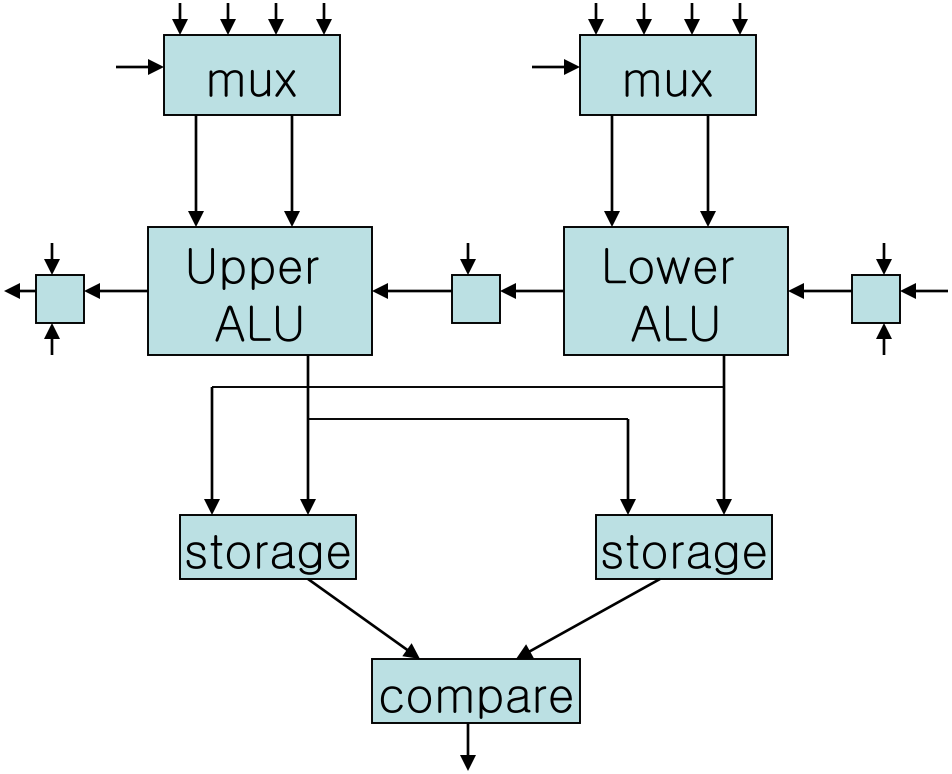
  [0, 2^i, 2^(i+1), 3x2^i]

- When two bits are shifted

  [0, 2^(i-2), 2^(i-1), 3x2^(i-2)]

Re-computing with Swapped Operands (RWSO)

- Concept
  - First: normal operation
  - Second: computation with swapped operands

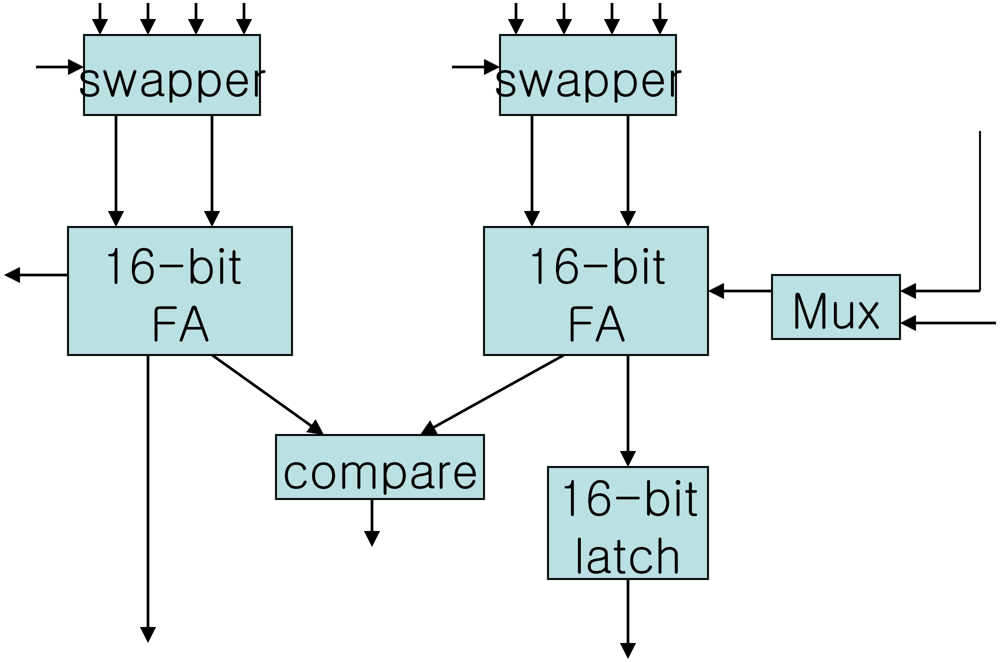


- When bit i is faulty,

  [0, 2^(i+r+1), 2^(i+r+2)]

Re-computing with Duplication with Comparison (REDWC)

- Concept
  - Error detection resembles that of duplication with comparison
  - First calculation: compute lower half using lower and upper part of ALUs and compare it.



Re-computing with Duplication with Comparison (REDWC)

- Characteristics
  - Same ability to detect a fault with duplication with comparison
  - Single fault is confined to one half of the adder

Re-computation for error-correction

- Concept
  - Repeated computation with shift
    - Can detect errors and locate the faulty unit
  - Only works for the independent unit
    - The adjacent bits are not independent.