**SIFT**

- SIFT = Software Implemented Fault Tolerance
  - Ultra reliable computer for critical aircraft control applications.
    - (desired probability of failure $< 10^{-10}/h$ over 10h mission)
  - Developed by SRI International for NASA Langley Research Center
  - Major goal was the application of formal verification of the critical functions
    - achieve fault-tolerance as much as possible by programs
  - Byzantine General Problem discovery
  - Objective for operating system was to hide the fault-tolerant mechanism from the application
  - Proof-of-concept demonstrator
    - $\Rightarrow$ bad news: 70-80% overhead (due to executives)
**SIFT**

- System Overview
  - structure of the SIFT system

Wen78 fig.1
**SIFT**

- **System Overview (cont.)**
  - loosely coupled multiprocessor system
    - implements message passing
    - no shared memory
  - up to 8 processors
    - Bendix BDX930 processors, 32kB RAM, $\lambda$ approx. $10^{-3}$
  - general processors and I/O processors
    - have private memory
    - I/O procs. have less RAM
  - complete interconnection
SIFT

- SIFT Operating System Functions
  - scheduling
    » OS must cause application tasks to be executed at proper time.
  - synchronization
    » OS must keep processors moderately synchronized (within 50µs)
  - consistency
    » OS must implement voting at input to process
  - communication
    » results of tasks must be transmitted to all processors in timely manner, i.e. shared results
**SIFT**

- fault masking
  - erroneous data, generated by faulty processor, must not be allowed to propagate, and source of erroneous data should be identified
  - => NMR

- reconfiguration
  - OS must be able to detect a faulty unit and reconfigure
  - => graceful degradation (hybrid NMR)

- input/output
  - OS controls received data from sensors and transmitted data to actuators
  - separate busses => sensors possibly act as faulty general
**SIFT**

- **Fault Isolation**
  - provided at the boundaries between processors and buses
  - processor can read memory of other processors, but can write only to its own memory.
  - a non-faulty processor can obtain bad data due to bad bus
  - invalid control signals should not produce incorrect behavior of non-faulty processors
**SIFT**

- **Fault Masking**
  - faulty units can provide processors with bad data
  - tasks place their outputs into a pre-broadcast buffer
  - this data is then broadcast to other processors
  - values from all processors are collected in the pre-vote buffer
  - the voted majority is then placed into the post-vote buffer
  - buffer management
    - 8 x 128 word data buffers
    - 1 buffer per processor
    - any message received from Processor $i$ goes to Buffer $i$
    - each message is tagged with ID (0,...127) => 128 shared variable names
SIFT buffer management and voting
**SIFT**

- **Voter**
  - Voting on exact match basis => bit-by-bit majority voting
  - If the voter cannot find a majority
    - a default value specified by the application is selected
  - After identifying faulty unit, system is reconfigured
  - The number of processors executing a task can vary with the task
    - dynamic configuration of task redundancy
    - takes advantage of the fact that in flight control particular tasks are more critical than others
**SIFT**

- **Scheduler**
  - outputs must be generated with specified frequency
  - delay between reading and outputting result must be bounded
  - table driven, making use of several independent schedules:
    - task schedule
      - list of tasks to execute, how often, and when.
    - Pre-vote schedule
      - list of buffers to be moved and data-file to the pre-vote buffer, how often they should be moved, and when.
    - voter schedule
      - list of buffers to be voted on, how often should be voted and when.
    - broadcast schedule
      - list of buffers to be broadcast, how often, and when.
    - copy schedule
      - list of buffers to be copied from one slot in the post-vote buffer to another.
SIFT

- Frame based, periodic scheduling
SIFT

- Scheduling strategies considered:
  » non-preemptive fixed priority scheduling
    ■ did not allow sufficient flexibility
  » priority scheduling
    ■ tasks with fastest iteration rate are given highest priority
    ■ works if utilization is below $\ln(2)$
  » earliest deadline first
    ■ not useful, since task durations are very short
  » simply periodic scheduling
    ■ all periods are multiples of smaller periods
- SIFT implements simple periodic scheduling with integer multiplicity of subframes.
SIFT

- Frame base scheduling
  » each task is assigned to one of several priority levels
  » each priority level corresponds to an iteration rate
  » each iteration rate is an integer multiple of the next lower one
  » subframes are 1.6 to 3.2 ms long
  » frames are up to 50 subframes long (< 160ms)
  » each task must fit into one subframe
    ■ left time is used for background diagnostic task
  » results are available in next subframe
  » schedule is table driven and static
  » at subframe boundary
    ■ suspend task
    ■ broadcast all data from last frame
    ■ invoke voting task
    ■ start scheduled task
Clock Synchronization

- observed scenario triggered Byzantine generals problem
  - Assume 3 clocks A, B, and C.
  - Assume that A runs slightly faster than B and that C has failed.
  - If C reports to A an even faster time and to B a slower time, then both A and B see 3 values, and will see the median value, which is their value.
  - Thus A and B will not change their clocks and will gradually drift apart.
SIFT

- SIFT requires $3N+1$ clocks to mask any $N$ simultaneous faults.

- Clock synchronization algorithm:
  - egocentric algorithm based on clock time messages
  - uses *interactive convergence* of Lamport

  » Each processor obtains the value of all other clocks remaining in the configuration, comparing those values to the value of its own clock to find the apparent skew between those clocks and its own.

  » The processor regards the skew of its own processor as zero.

  » The skews are examined to find any that are greater than some threshold. Such skews are set to zero.

  » The arithmetic mean of the skews is calculated and the mean is used to correct the processors own clock.
SIFT

- Reliability Modeling
  - Markov Model
    - transitions caused by
      - fault occurrence
      - fault detection
      - fault handling
    - hardware fault, transient faults assume exponential distributions
      - executive considers unit faulty only after multiple transient faults
      - fail rates: processors $\lambda=10^{-4}$, busses $\lambda=10^{-5}$ per hour
    - state labeled by triplet $(h,d,f)$ with $h \leq d \leq f$
      - $f =$ number of failures occurred
      - $d =$ number of failures detected
      - $h =$ number of detected failures that have been handled by reconfiguration
SIFT

- Markov model
  - exhaustion of spares
    - $F = 5 \times 10^{-12}$
  - double faults ($t=100\text{ms}$)
    - $F = 5 \times 10^{-11}$
  - double fault ($t=1\text{s}$)
    - $F = 5 \times 10^{-10}$

Wen78 fig. 3
**SIFT**

- **Software**
- **Application Software**
  - performs actual flight control operations
  - fact that tasks are executed redundantly on different processors is transparent to application
  - set of iterative tasks
  - in each iteration, an application task gets its input from a calls to the executive
  - results are made available as inputs to next iteration tasks by executing calls to executive
  - inputs or outputs consist of few words of data
Executive Software

- Responsible for the reliable execution of the application tasks
  - run each task at the required iteration rate
  - provide correct input values
  - detect errors, diagnose their cause
  - reconfigure system after failure

- 3 Executive Components
  - global executive task
    - executed as critical task => TMR using majority voting
  - local executive
    - local task execution issues
  - local-global communication tasks
    - reconfiguration & error reporting
SIFT

- logical structure of SIFT software system

Wen78 fig. 5
SIFT

- arrangement of application tasks within SIFT configuration

Wen78 fig. 6
**SIFT**

- arrangement of executive within SIFT configuration

Wen78 fig. 7
- **Local-Global Communication Tasks**
  » Consists of reconfiguration task and error reporting task
    - restatement of Wen78 fig.8
      - Error handler in each processor puts reports in error table
      - Error reporter task in each processor reads error table and decides what conditions to report to the global executive. This report is put in a buffer.
      - Global executive (triplicated) reads each processor’s buffer over three busses (to guard against bus errors) and votes for a plurality.
      - Global executive, using the diagnosis provided by the error reporter, determines what reconfigurations, if any, is necessary. If a reconfiguration is necessary, a report is put in a buffer.
      - Local reconfiguration task in each processor reads report from each of the global executive buffers and votes to determine plurality.
      - Local reconfiguration task changes the scheduling table to reflect the global executive’s wishes.
SIFT

» error reporter
  ■ keeps error count on all processors
  ■ considers processor faulty when error count exceeds some threshold
  ■ periodically broadcasts error report

» reconfiguration task
  ■ votes on reconfiguration recommendation from the global executive OM(1)
  ■ adjusts pre-computed schedules and buffer tables
  ■ for an 8 processor SIFT there are 8 sets of tables
  ■ processors have virtual processor numbers, i.e. given a particular combination of non-faulty processors, each processor needs to determine its virtual proc. #
    ■ the processor uses this number to decide which task it should be running
  ■ reconfiguring means: determine which set of tables to use, mapping real processor numbers to virtual numbers, building schedule tables.
- **Global Executive**
  - executed as critical task => TMR using majority voting
  - analyzes error reports, diagnoses errors
  - determines failed processor: whenever 2 or more processors identify another processor as malfunctioning, recommend reconfiguration.
  - list of processors recommended to be reconfigured out of the system is broadcast to reconfiguration task.
**SIFT**

- **Local Executive**
  - collection of routines
  - error handler
    - invoked by voter in case of fault detection
    - updates error table
  - scheduler
    - schedule execution of tasks
    - use predefined iteration rate
    - define sequence of time frames
    - whereas execution must occur within a frame, it could be at any time during that frame
    - scheduler invoked by timer interrupt or completion of task
    - tasks may be preempted by clock, e.g. in order to run task with higher priority
SIFT

» buffer interface routines
  ■ double buffering: input buffer contains value from last iteration that will be used as input to task.
  ■ in next frame buffers are switched
» voter routine
  ■ invoked by a task to get inputs for its current iteration
  ■ voter uses table of reconfiguration task to determine which processor and which buffer contains copies of the output to be voted on.
  ■ if disagreement, invoke error reporter.

Wen78 fig. 9
SIFT

- Fault Detection
  » analysis of faults to determine faulty component
  » each processor maintains local processor/bus m-by-n matrix.
    \( m = \# \text{ processors}, \ n = \# \text{ busses} \)
  » entry \( X_{p[i,j]} \) in processor \( p \)'s table is the \# of faults detected by
    local executive involving processor \( i \) and bus \( j \).
  » assume processor \( p \) is reading from processor \( q \) using bus \( r \)
    ■ interface from \( r \) to \( q \) is faulty
    ■ interface from \( p \) to \( r \) is faulty
    ■ bus \( r \) is faulty
    ■ proc. \( q \) is faulty
    ■ proc. \( p \) is faulty