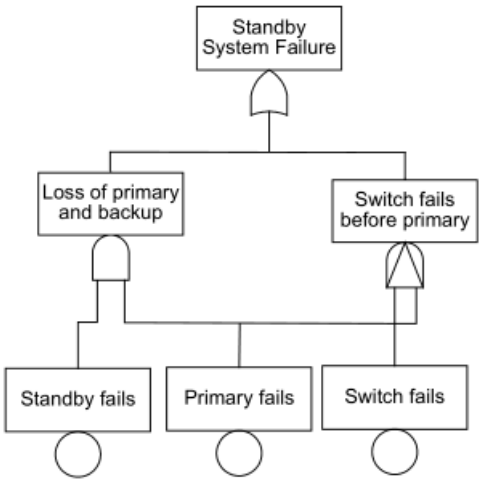
FTA reminders

- **Define top event** (Loss of System OR Loss of Mission)
- **Define scope** (version, initial states of components, assumed inputs to SUT)
- **Define resolution** (what is too low-level and will be excluded)
- **Think small**: "necessary and sufficient IMMEDIATE events" = smallest possible steps, don't jump to basic causes
- **Do not mix successes with faults**
- **For each event, state exactly**
  - What the faults is
  - When it occurs
- For each step, identify **all possibly related faults**, not just obvious ones:
  - component failed itself
  - component failed in the result of upstream fault(s)
  - component failed in the result of external influence
- **Fault nodes**:
  - **Component fault**: uniquely associated with 1 component
  - **System fault**: caused by
    - more than 1 component
    - external factor
- **For each component fault**, identify:
  - **Primary fault**: should work but fails
  - **Secondary fault**: fails because environment outside design scope
  - **Command fault**: does what is designed to but *when* it shouldn't
- **For each system fault**, identify:
  - **(minimum necessary** & **sufficient) immediate** causes
  - **INHIBIT** gates (outside factors)
- **No saving miracles** rule:
  When something could stop failure propagation if it failed or worked not-as-designed, don't assume it would

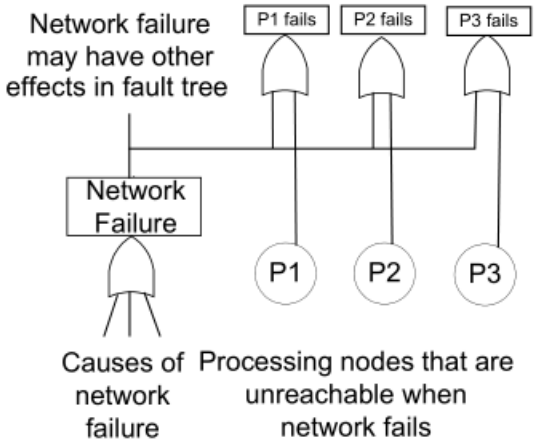
Dynamic FTA

use Priority AND to analyze time dependencies (if FIRST that and THEN that happens):



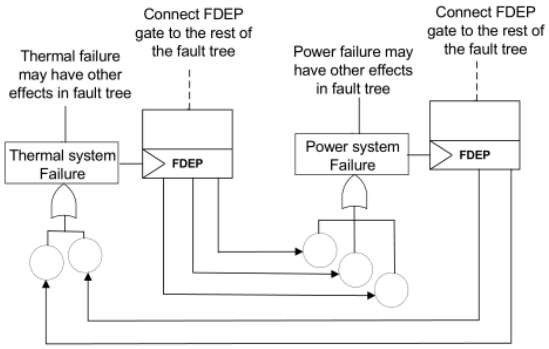
Functional dependency

FDEP box or shared input:



Feedback loops

FDEP box

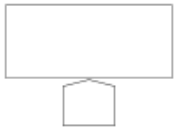


PRIMARY EVENT SYMBOLS

BASIC EVENT - A basic initiating fault requiring no further development

CONDITIONING EVENT - Specific conditions or restrictions that apply to any logic gate (used primarily with PRIORITY AND and INHIBIT gates)

UNDEVELOPED EVENT - An event which is not further developed either because it is of insufficient consequence or because information is unavailable



HOUSE EVENT - An event which is normally expected to occur

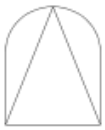
GATE SYMBOLS

AND - Output fault occurs if all of the input faults occur

OR - Output fault occurs if a least one of the input faults occurs

COMBINATION - Output fault occurs if n of the input faults occur

EXCLUSIVE OR - Output fault occurs if exactly one of the input faults occurs



PRIORITY AND - Output fault occurs if all of the input faults occur in a specific sequence (the sequence is represented by a CONDITIONING EVENT drawn to the right of the gate)

INHIBIT - Output fault occurs if the (single) input fault occurs in the presence of an enabling condition (the enabling condition is represented by a CONDTIONING EVENT drawn to the right of the gate)

TRANSFER SYMBOLS

TRANSFER IN - Indicates that the tree is developed further at the occurrence of the corresponding TRANSFER OUT (e.g., on another page)

TRANSFER OUT - Indicates that this portion of the tree must be attached at the corresponding TRANSFER IN