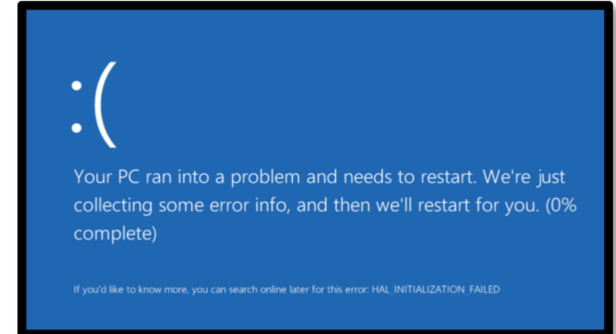
Fault Resilient Voltage Scaling of Embedded Memories for Efficient Operation at the Edge

Irina Alam and Puneet Gupta

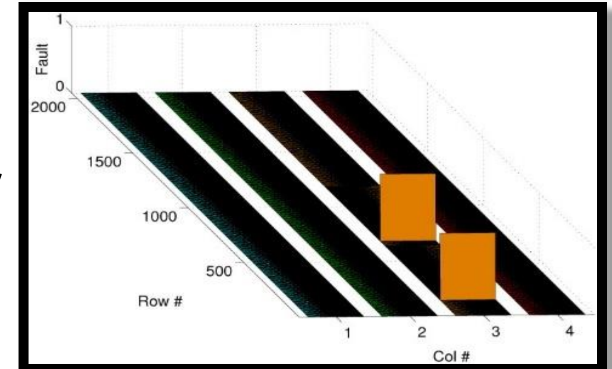
Department of Electrical and Computer Engineering,
University of California, Los Angeles

Motivation: Memory Errors are a Major Problem

- **DRAM reliability worsens** with density
  - Google: 70,000 FIT/Mb in commodity DRAM; 8% of modules affected per year;
    4% of servers crash per year [Schroeder CACM'11]
  - Facebook: 2.5% of machines see DRAM errors per month [Meza DSN'15]
- **SRAM stops working** at low voltage
  - 6X fault rate measured from 600mV to 525mV [Gottscho TACO'15]
- **Flash wears out** with usage
  - NASA's Opportunity Mars rover had to reformat its flash in 2014
- **STT-RAM is unpredictable**
  - Stochastic write & thermal instability [Zhao Microelec. Rel.'12]

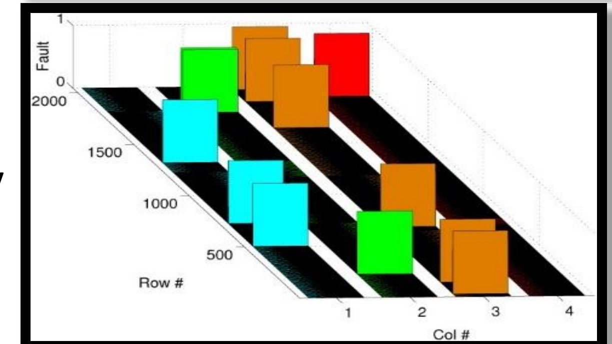


550 mV



Memory errors will continue to be a challenge!

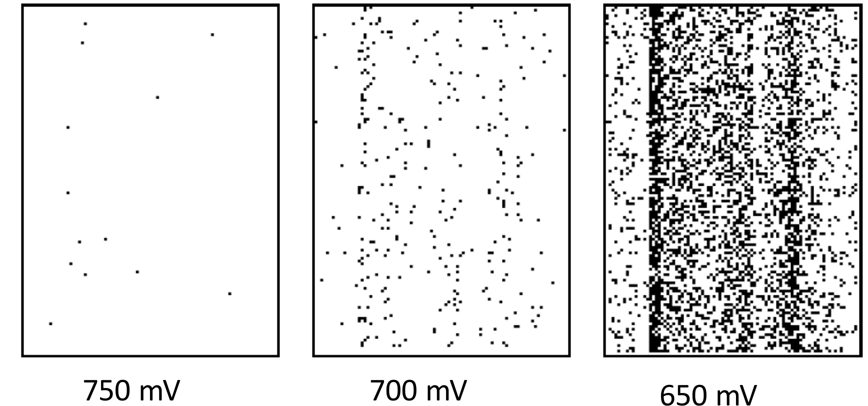
525 mV



[Gottscho TACO'15]

Embedded Memory in IoT Devices

- **Low cost and low energy consumption**
- **On-chip memory consumes significant system energy**
  - Possible solution – reduce supply voltage
    - *Hard fault rate rises* exponentially
    - *More susceptible* to radiation induced soft faults



- **Memory Fault Tolerance**
  - **Virtualization-free : No OS support**
    - **No memory reliability techniques (e.g. Page Retirement)**
  - **Large voltage guardbands – reduces battery life**
  - **ECC schemes such as SECDED/Chipkill - too expensive for IoTs**

High Reliability, alongside Low Cost and Low Energy for embedded memory - still a Challenge!

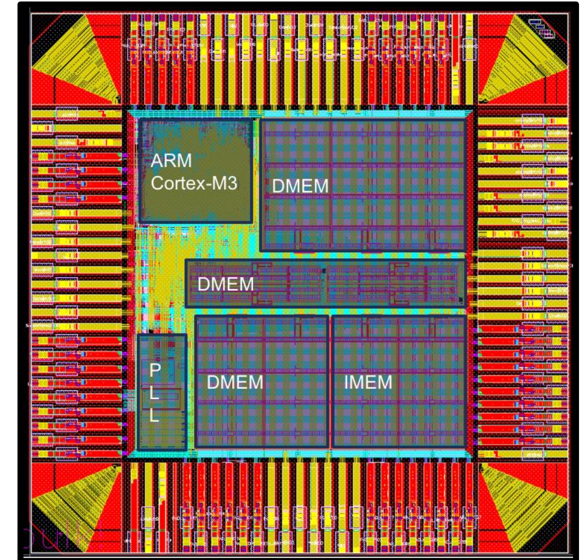
Question

How to achieve **Low Cost** *Memory Fault Tolerance* for "embedded systems at the edge of the Internet-of-Things" with **minimal hardware changes**?

FaultLink

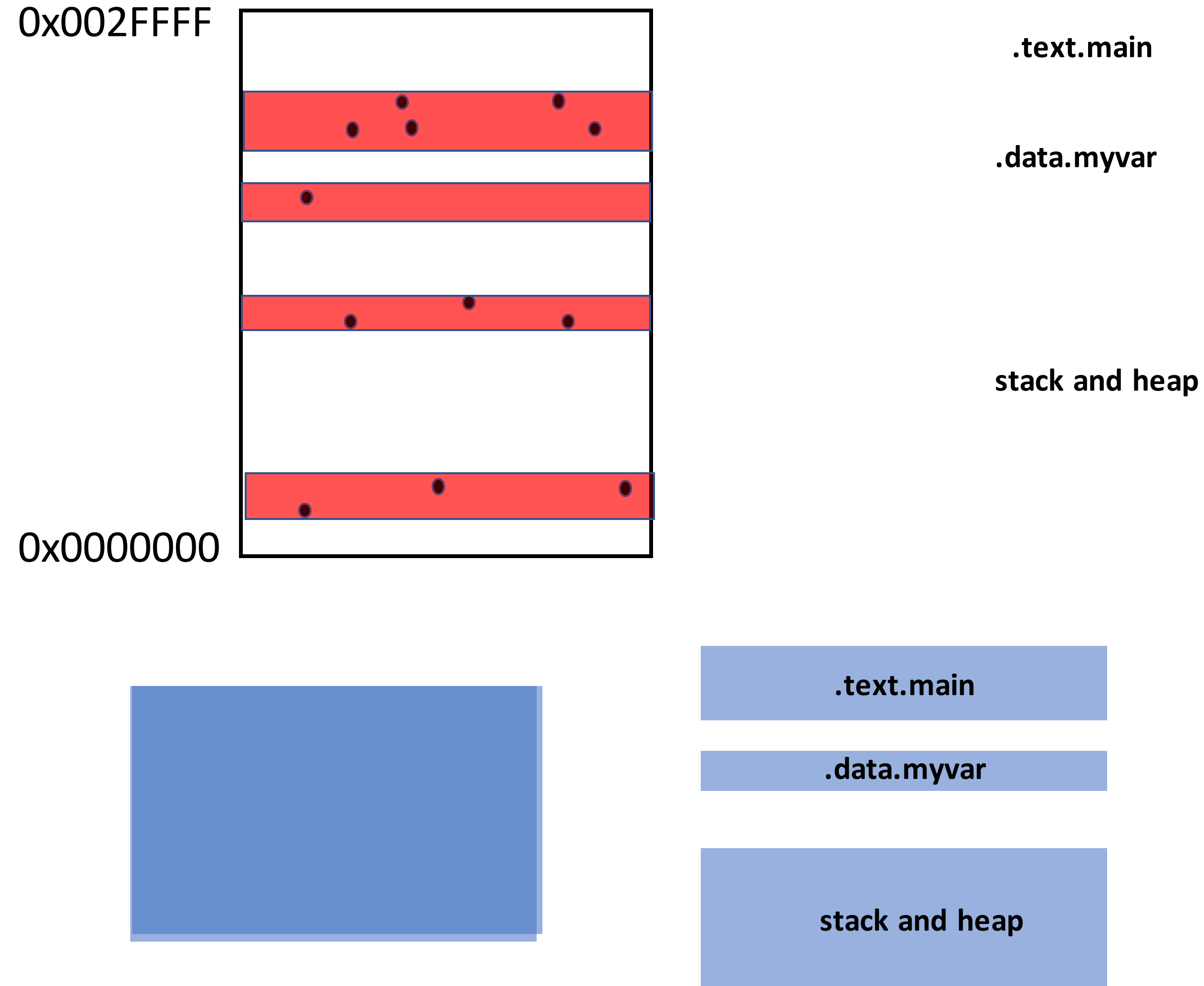
FaultLink: Guarding Against Hard Faults at Link Time

- Virtualization-free *software-managed* scratchpad memories
- <u>FaultLink</u> - Tolerating hard faults:
  - Allows loading application into faulty memory
  - Can run application at lower voltage
  - At build time, avoid accessing bad memory locations
  - Each chip gets unique binary image: link many times



How FaultLink Works?

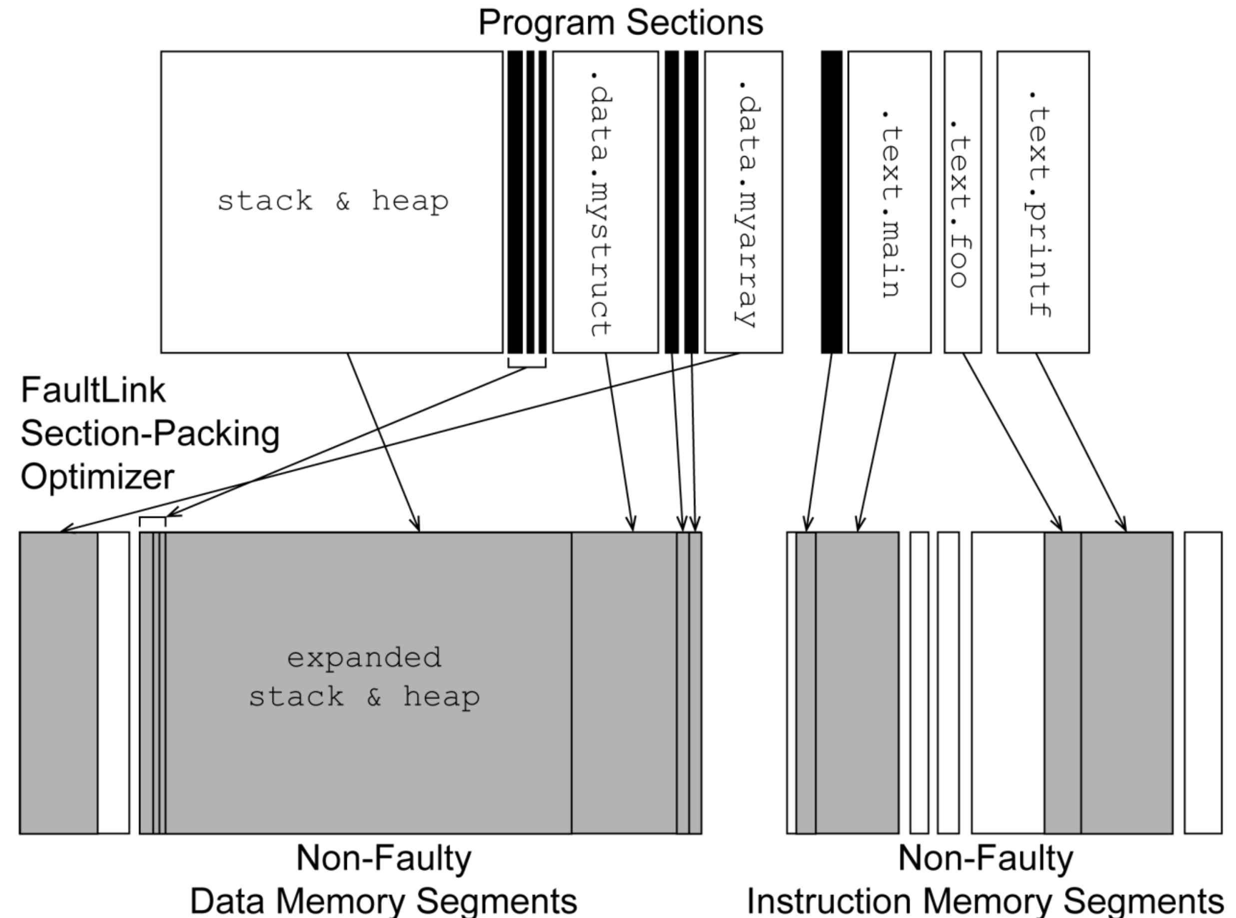
- Scratchpad memory – software managed
- Memory chip starts seeing faults - application fails
- Characterize memory - Identify fault locations
- Early app. compile with special flags
- Just-in-time link



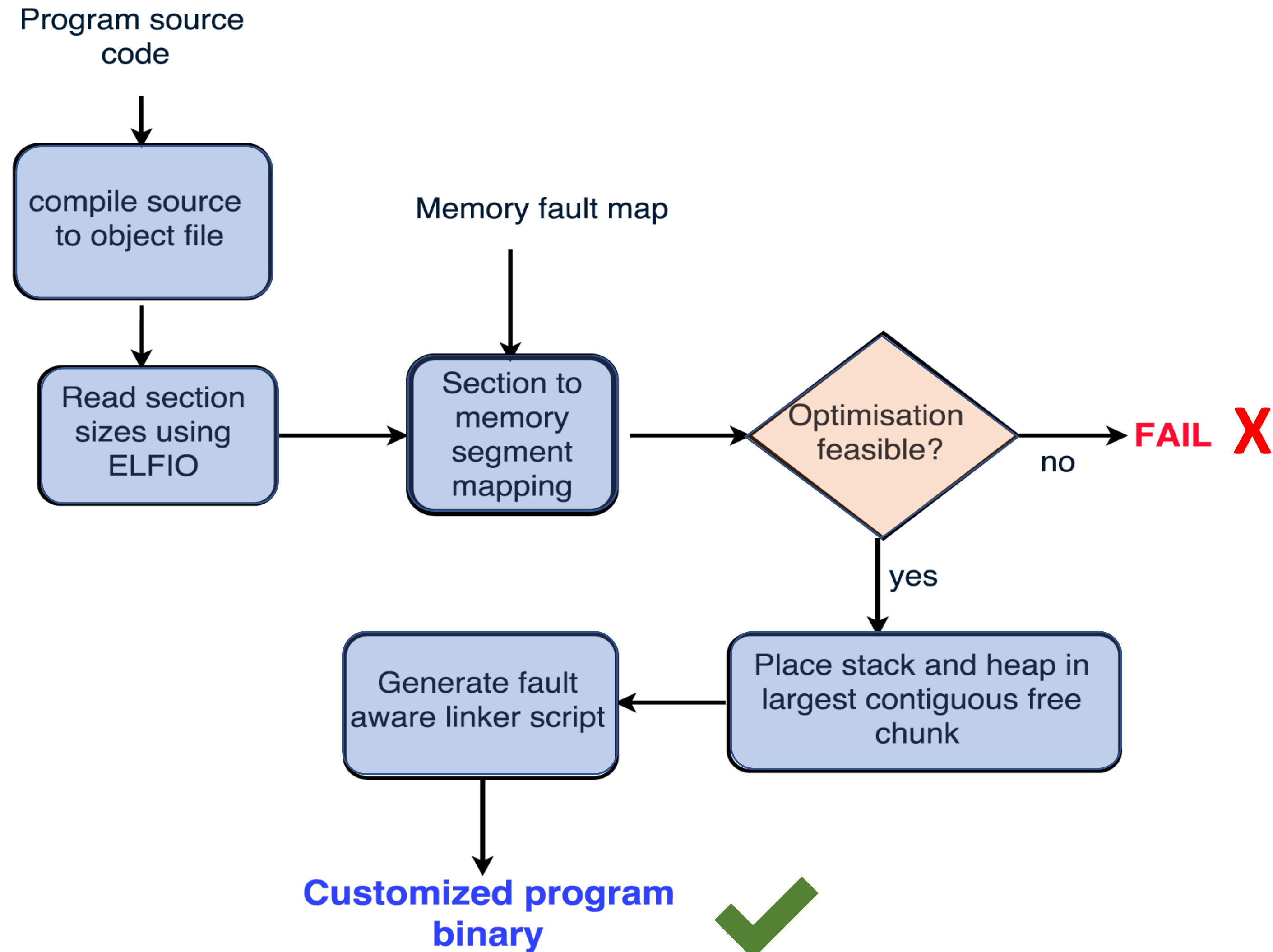
FaultLink Problem Formulation

ILP problem formulation

- Variant of the Multiple Knapsack Problem
- Objective: minimize # of packed segments
- Constraints:
  - Every program section packed in exactly one contiguous fault-free memory segment
  - Total size of packed program sections in a memory segment <= segment size



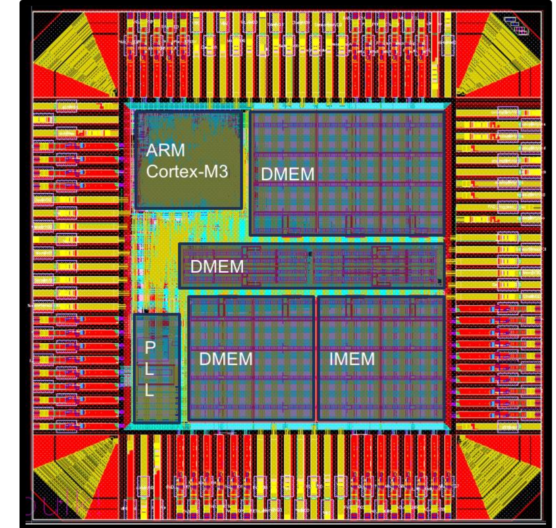
The FaultLink Flow



Evaluation Setup

- *Real* Test Chips
  - 45nm Microcontroller-Class Test Chips : 8
  - I-MEM: 64KB; D-MEM: 176KB
  - Injected faults : Lowering VDD from 1V to 600mV in 50mV increments
  - Benchmarks loaded and ran at each VDD till completion (if possible)
  - Verified the binary on simulator to check if faulty locations are accessed

  - Small embedded-class benchmarks used –
    - MiBench : blowfish, sha
    - Others: dhrystone, whetstone, integer matrix multiplication



Evaluation Setup

- *Synthetic* Test Chips
  - Instruction and data SPM capacities – 128KB, 256KB, 512KB, 1MB, 2MB, 4MB
  - Synthesized 100 fault maps in 10 mV increments
    - Used Monte Carlo simulations of SRAM bit-cell noise margins in the 45nm technology
  - Compiled for the open-source 64-bit RISC-V (RV64G) instruction set
  - FaultLink produced binary ran till completion on simulator (if possible)

  - Six additional bigger benchmarks added for evaluation:
    - Approximation-friendly AxBench suite – blackscholes, sobel, jpeg, jmeint, fft, inversek2j

FaultLink Results : Reduction in Supply Voltage

- **Measured min Vdd Reduction on Real Test Chips**
  - **Reduction of 300mV** compared to the official technology specification of 1V
  - **Reduction of 125mV** compared to the average non-faulty min-VDD of 825mV

- **Yield at min-VDD for Synthetic Test Chips**
  - At 99% yield, **min-VDD is reduced by up to 450mV** with respect to nominal 1V
  - **Reduces between 370mV and 430mV** with respect to the lowest non-faulty voltage

DL - FaultLink

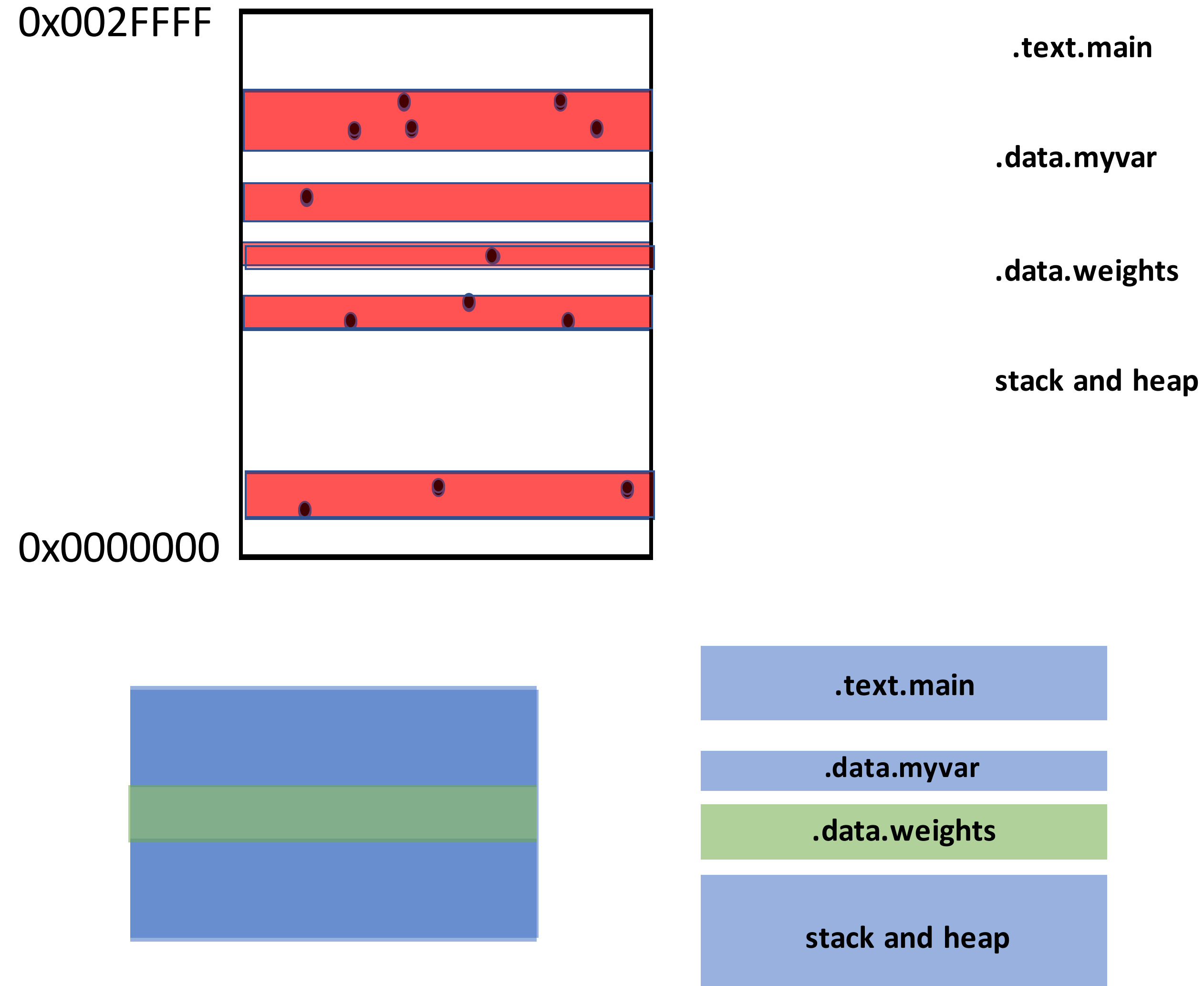
FaultLink for ML workloads

Fault Resilient DL networks

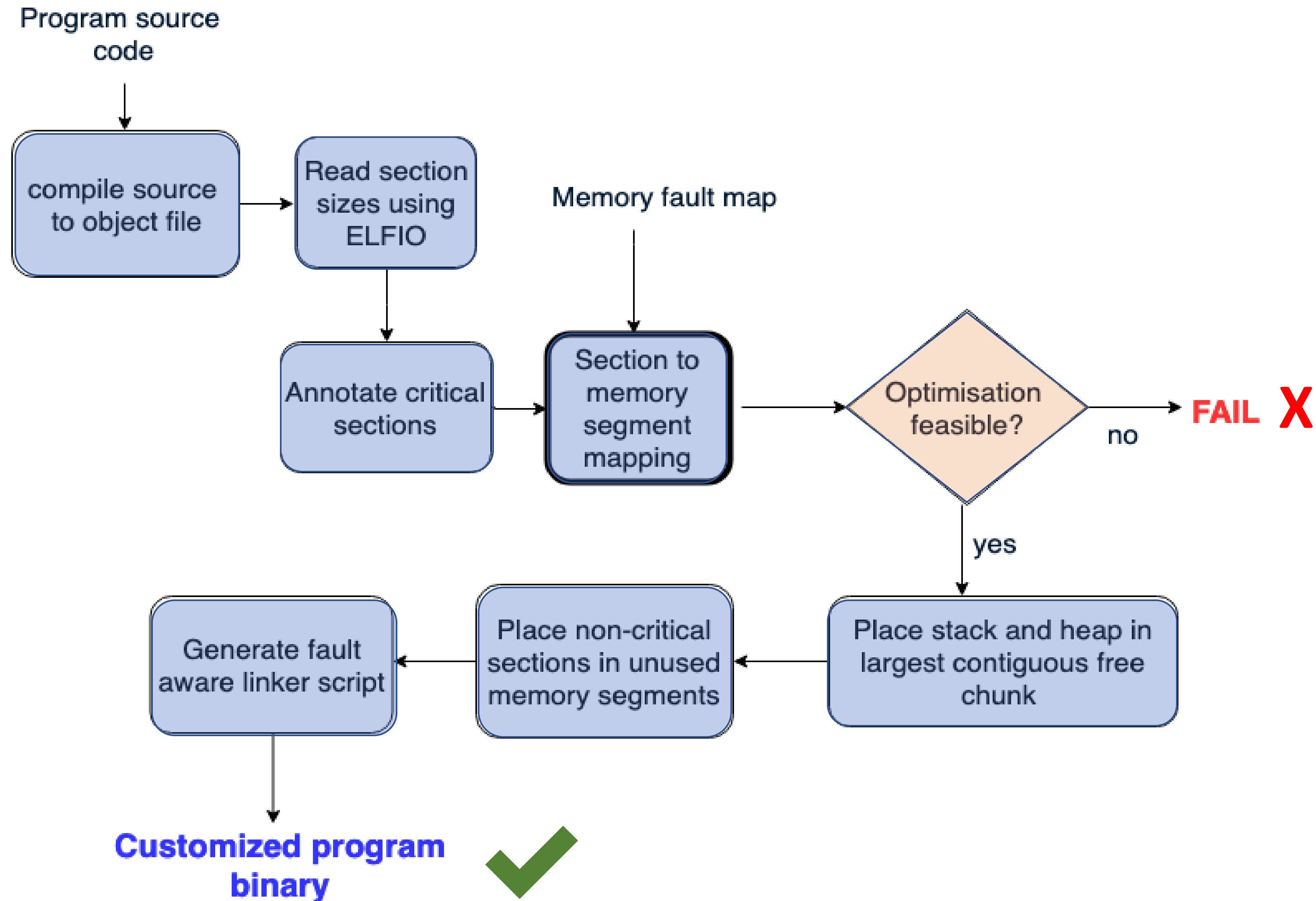
- Abundant redundancy in Deep Learning Neural Networks
  - Moderately fault resilient
  - Allows significant compression without impacting accuracy
- Resilience of DL networks depend on:
  - Type of data (inputs vs weights)
  - Data values/ data types
  - Layer type/ position in network (input layer vs hidden layer)
- Code and data in DL networks can be split into:
  - Critical
  - Non-critical

How DL - FaultLink Works?

- Scratchpad memory – software managed
- Memory chip starts seeing faults - accuracy drops
- Characterize memory - Identify fault locations
- **Split into critical and non-critical sections**
- **Link critical sections**
- **Link non-critical sections**



The DL-FaultLink Flow



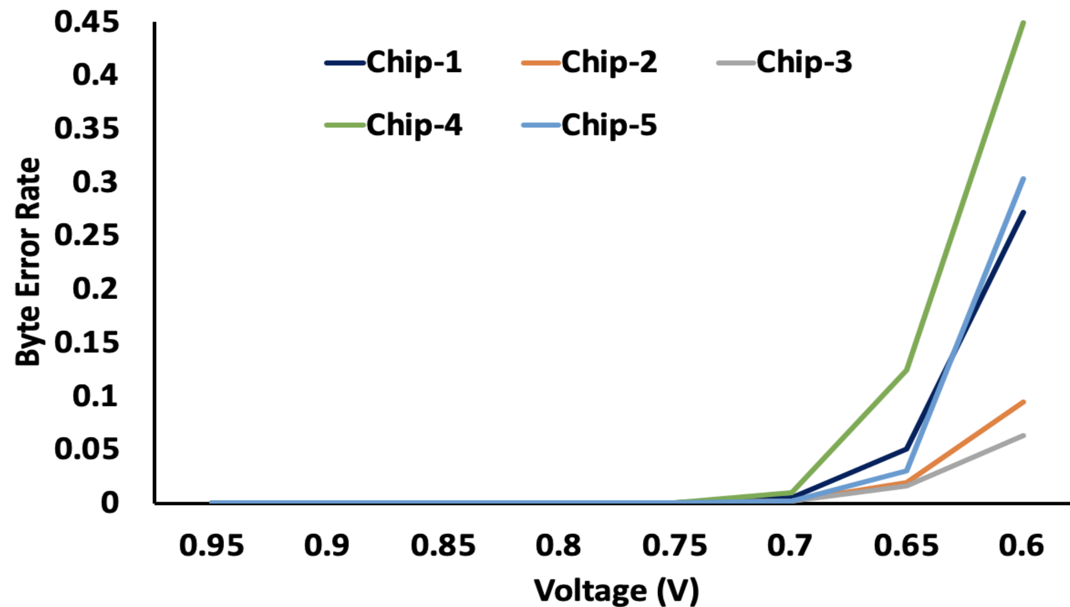
Networks evaluated

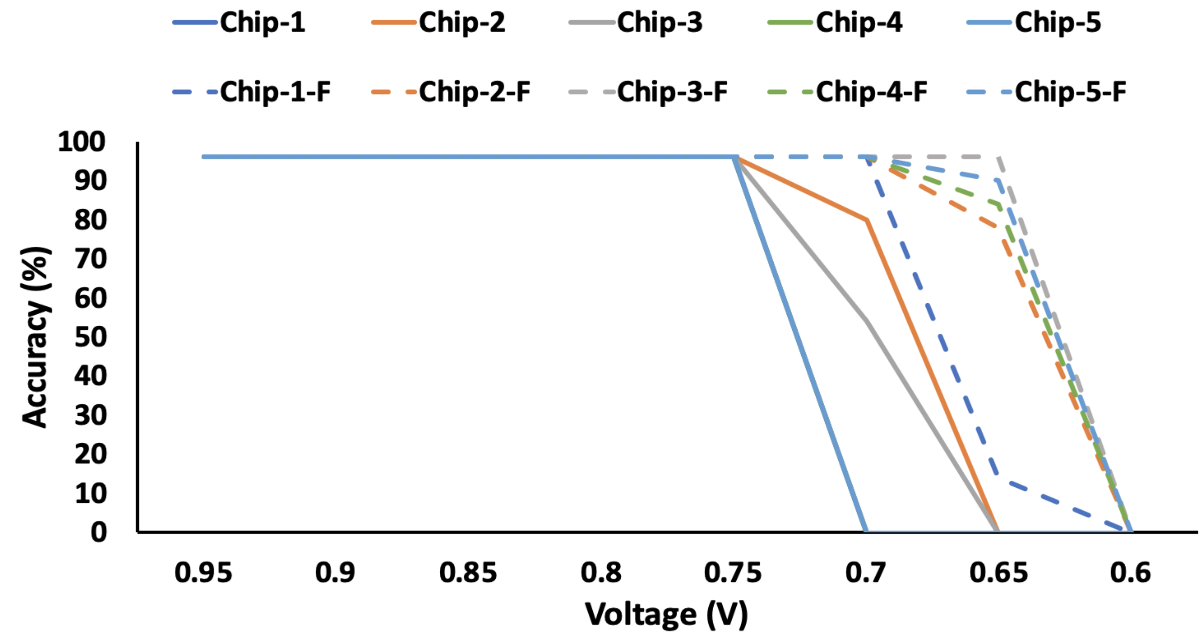
- DL *binarized* networks evaluated (on MNIST) –
  - MLP : 784-128-10
    - Minimally sized MLP with one hidden layer
  - CNN : 32CONV5-MP2-32CONV5-MP2-10FC
    - Modified version of LeNet [1]

- Binarized vs. 8-bit precision –
  - Loss in accuracy:
    - MLP: ~5%
    - CNN: ~1.6%
  - Network size
    - MLP – 7.8x smaller
    - CNN – 8x smaller

[1] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: Proceedings of the IEEE 86.11 (1998), pp. 2278–2324.

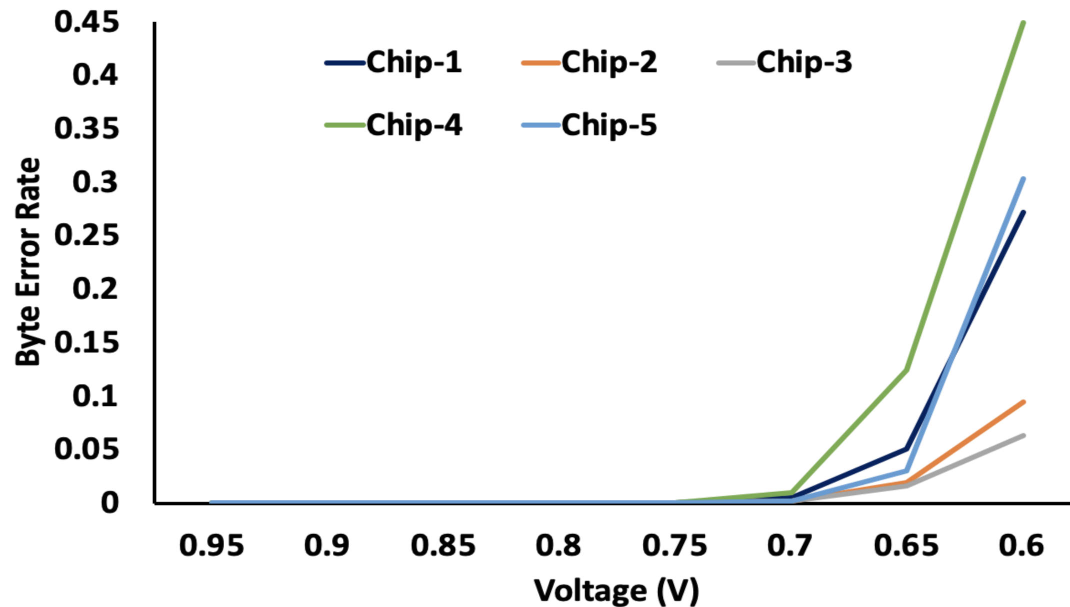
Results - MLP

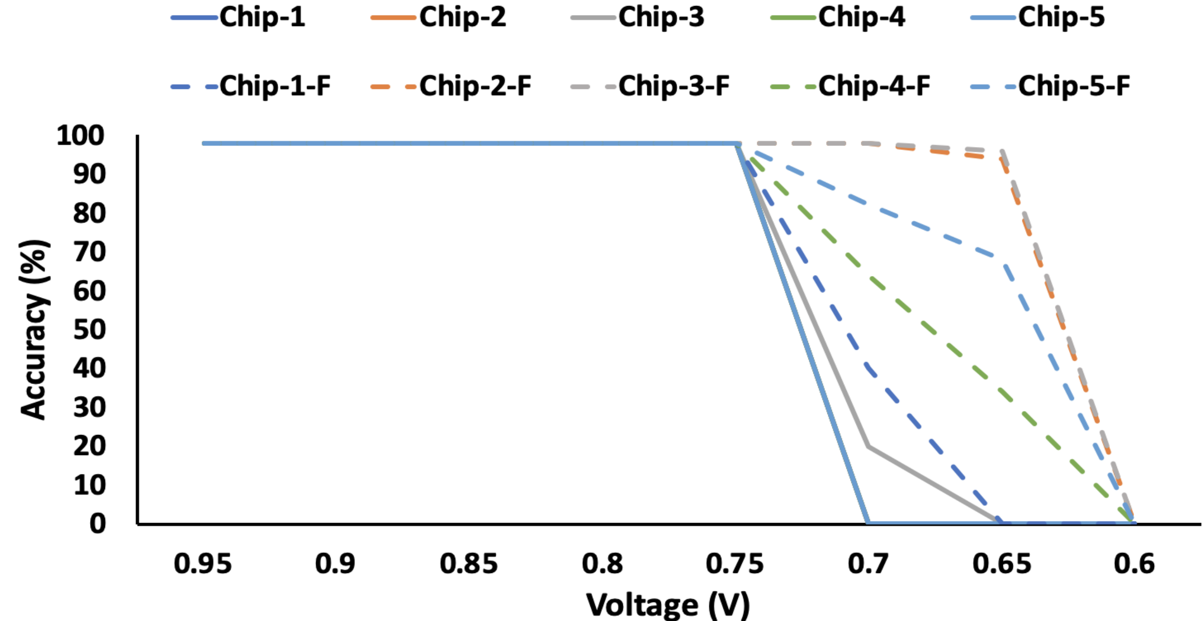




- Without DL-FaultLink : 750mV
- With DL-FaultLink : 100mV reduction from 750mV with no impact on accuracy
  - ***At 650mV***
    - ***Chip 1 – Accuracy drops to 14%***
    - ***Chips 2, 3, 4 and 5 – No/minimal impact on network accuracy***

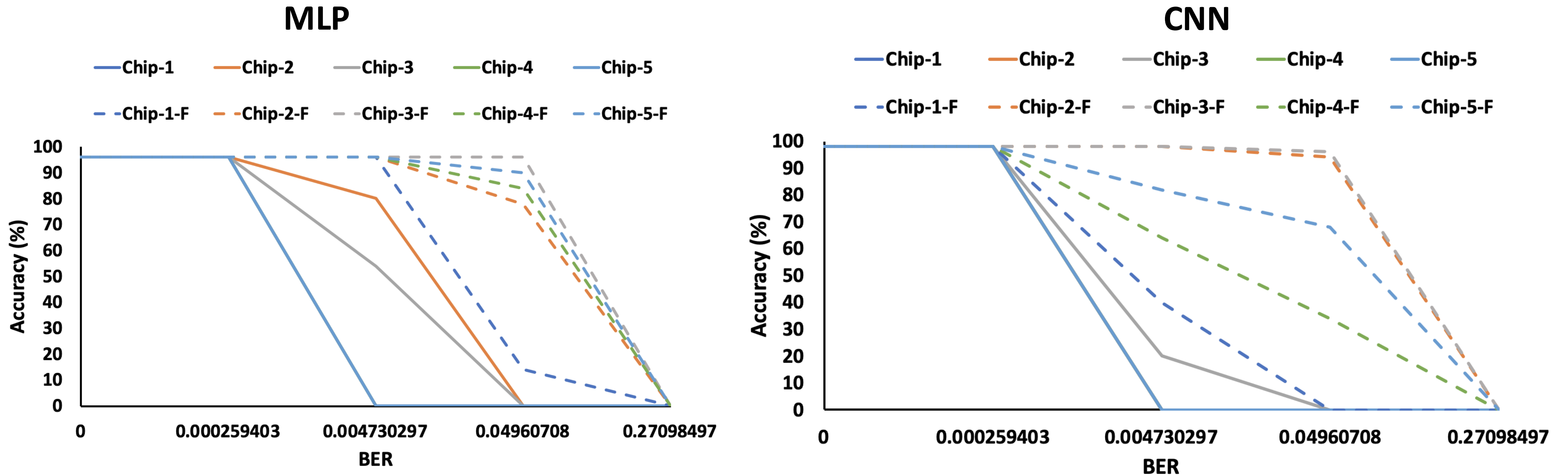
Results - CNN





- Without DL-FaultLink : 750mV
- With DL-FaultLink : Upto 100mV reduction from 750mV with no impact on accuracy
  - ***At 650mV***
    - ***Chip 1 – Packing failed***
    - ***Chips 2 and 3 – No/minimal impact on network accuracy***
    - ***Chips 4 and 5 – Accuracy drops to 34% and 68% respectively***

Fault Tolerance Technique

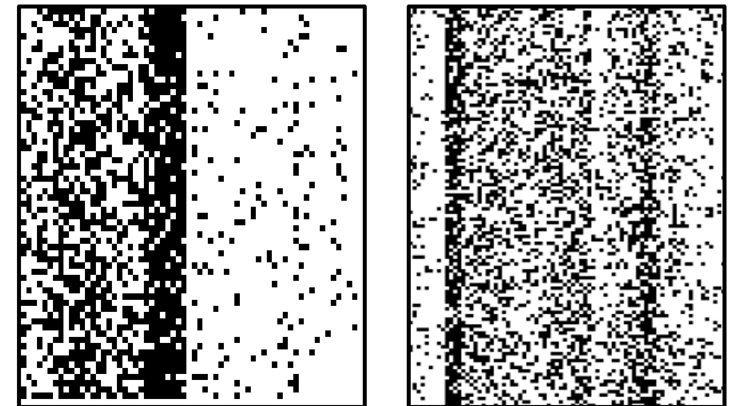


- DL-FaultLink: **<u>Aging induced fault tolerance technique</u>**
- Tolerates ~80x(avg) higher fault rate with NO impact on accuracy

Work in Progress

- Support for sparse networks
  - Lesser redundancy in weight tensors
  - Fragmented weight tensor for more placement opportunities

- Fine-grained error intersection
  - Map non-critical sections such that non-critical bits intersect with faulty location
    - *Bit-wise fault map instead of byte-wise fault map*
  - Utilize Significant Correlation in Fault Locations



Thank You!

Backup

Analytical FaultLink Yield Estimation

- Size of maximum contiguous program section comprises a significant portion of the overall program size
- Most section-packing failures occur when the largest section is larger than all non-faulty memory segments

Analytical Equation

L_k: RV, longest run of heads in k flips of a biased (p) coin.

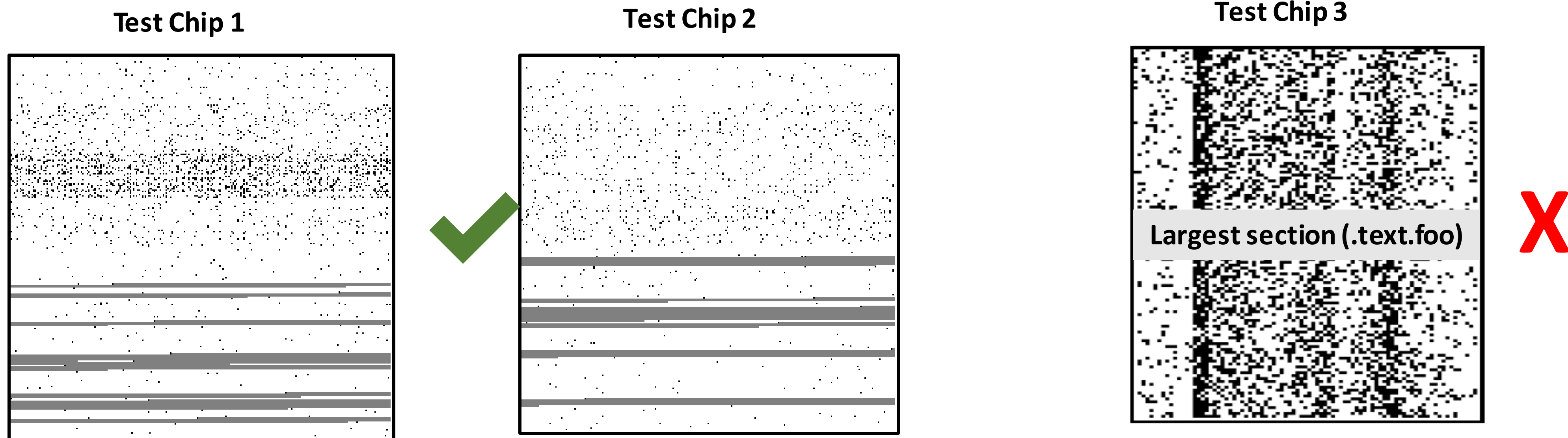
$$P(L_k < x) \approx e^{-p^{\left(x - \log_{p^{-1}}(k(1-p))\right)}}$$

Now we convert to our problem:

$s = (1-b)^{32}$

$$P\left(L_{size/4} < \frac{m_{max}}{4}\right) \approx e^{-s^{\left(\frac{m_{max}}{4} - \log_{s^{-1}}\left(\frac{size}{4}(1-s)\right)\right)}}$$

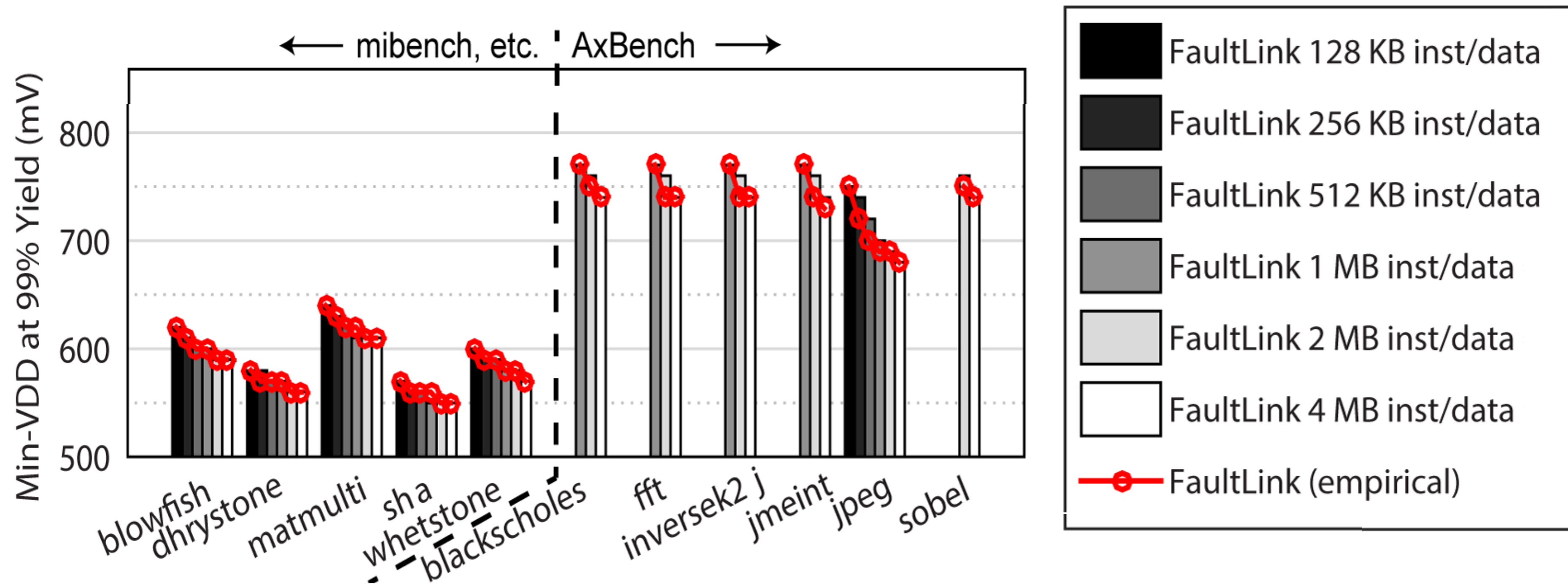
FaultLink Results : Tolerating Hard Faults



- **FaultLink successfully packed applications in faulty memory chips**
- **Packing failed mostly when there was no space for the biggest program section**

  → Analytical model can predict packing failures accurately

FaultLink Results: Impact of Memory Size

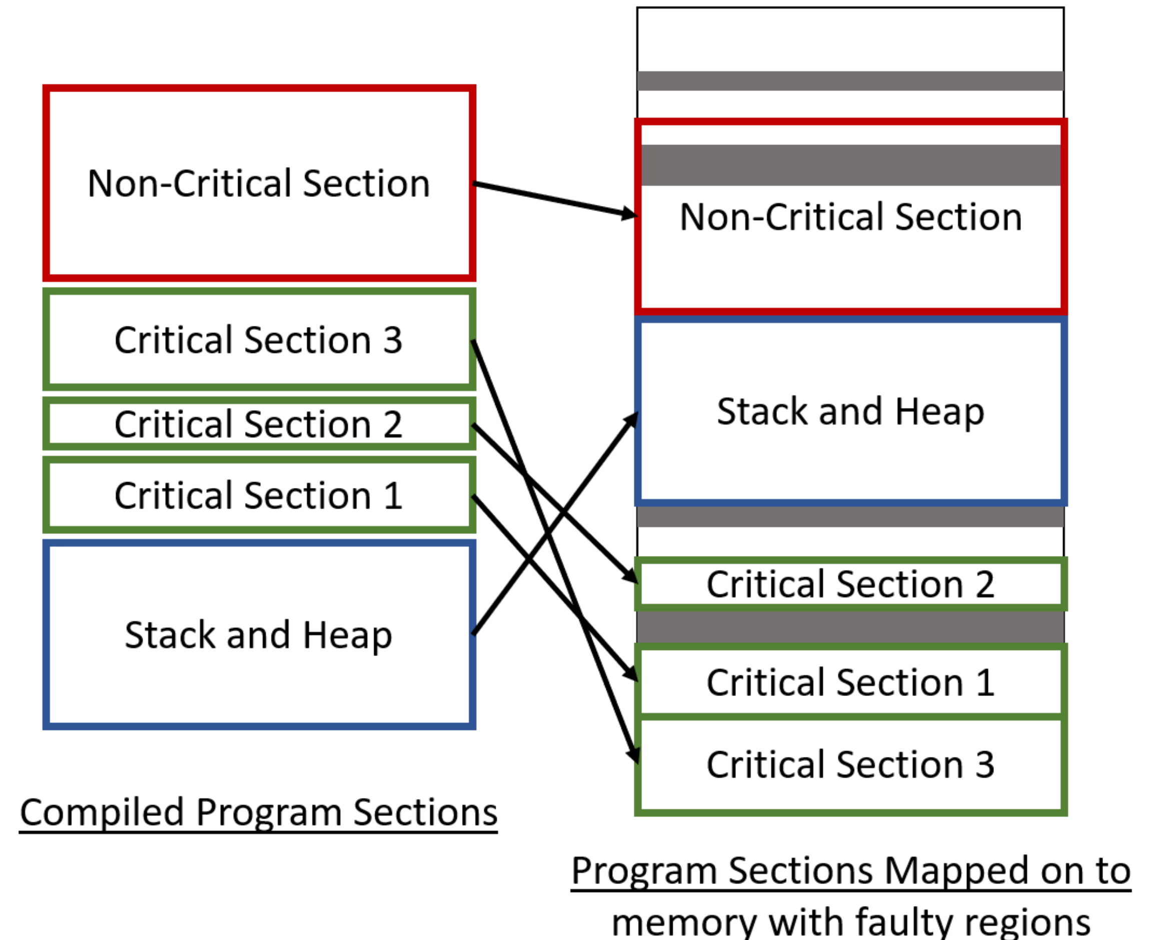


- **Benefit from increasing the size of memory saturates**
  - **Memory faults are independent and identically distributed (i.i.d.)**
  - **FaultLink failures are from large section packing failures**

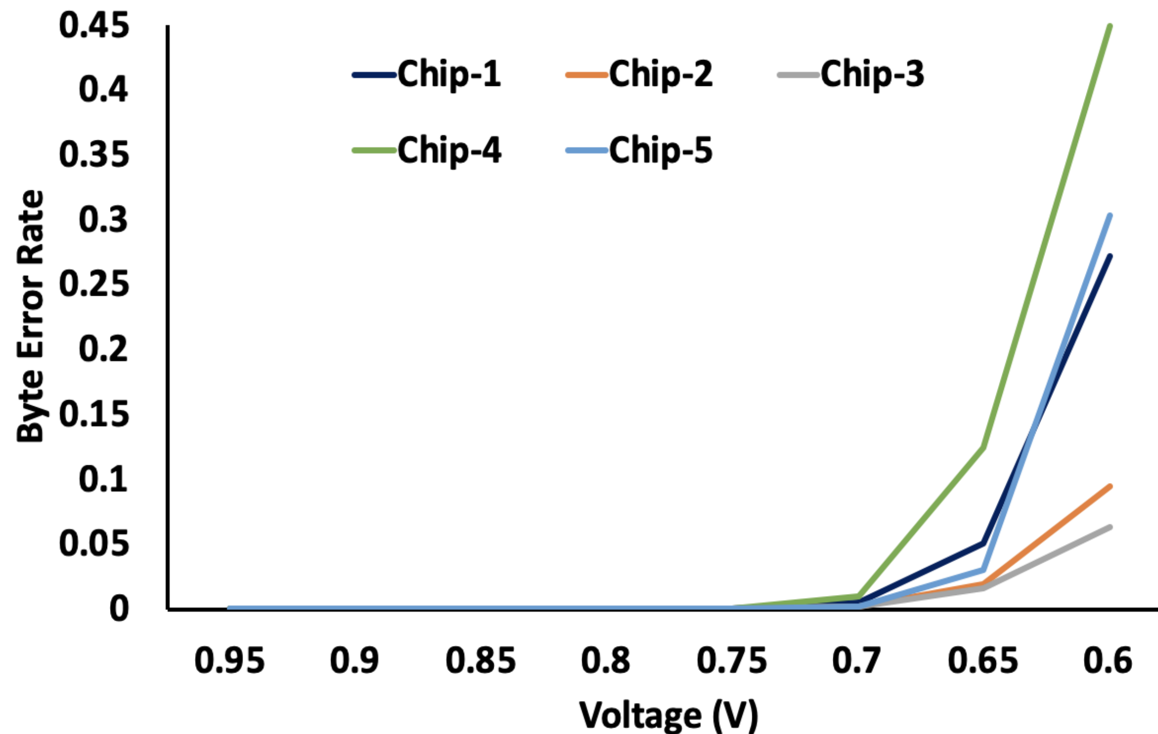
DL-FaultLink Problem Formulation

ILP problem formulation (for critical program sections)

- Variant of the Multiple Knapsack Problem
- Objective: minimize # of packed segments
- Constraints:
  - Every critical program section packed in exactly one contiguous fault-free memory segment
  - Total size of packed critical program sections in a memory segment <= segment size



Results – Error Rate



- First faults start appearing around 800mV
- Fault rate increases exponentially beyond 750mV
  - ***25x increase in fault rate between 750mV and 700mV***

Work in Progress

- Identify critical and non-critical sections for larger networks
  - Layer-type/position
  - Inputs/weights – specific values/bit positions
  - Three different placements of weight matrices
    - CNN - 10%, 40%, 98% accuracy for same chip at same voltage

0	1	0	0	1	1	1	1
1	1	0	1	0	0	0	1
0	1	0	0	1	0	1	0
0	0	1	1	1	0	0	0
0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0

0	1	0	0	1	1	1	1
1	1	0	1	0	0	0	1
0	1	0	0	1	0	1	0
0	0	1	1	1	0	0	0
0	0	0	0	0	1	0	0
1	1	0	0	1	0	0	0

- Intelligent placement of non-critical program sections
  - Map non-critical sections such that non-critical bits intersect with faulty location
    - *Bit-wise fault map instead of byte-wise fault map*
  - Significant Correlation in Fault Locations